HITEC ROBOTICS

ROBONOVA-I

Manual de Instrucción

ROBONOVA-I Manual de instrucciones

Tabla de contenidos

Especificaciones de ROBONOVA-I - 3

I. Antes de comenzar
1. Notas sobre el montaje y operatoria - 4
2. Seguridad al manejar baterías - 5

II. Notas sobre el montaje
1. Lista de partes - 6
2. Notas sobre servos y soportes - 8
3. Herramientas necesarias para el montaje - 11
4. Especificaciones del HSR-8498HB - 11

III. Montaje del kit
1. Ajustes de los horns de los servos - 12
2. Como poner el soporte a un servo - 14
3. Montaje de las piernas - 16
4. Montaje de los brazos - 28
5. Montaje del cuerpo - 30
6. Conectando las piernas y los brazos al cuerpo - 35
7. Conectar la cabeza al cuerpo - 37
8. Poner la tapa frontal del cuerpo - 38
9. Colocar el controlador en el robot y comprobar los recorridos de los servos - 39
10 Conexiones de los cables de los servos - 41
11. Montaje final del hardware - 49
12. Instalación de remocon y del sensor IR - 52

IV. Preparativos antes de manejar el ROBONOVA-I
I. Configuración de movimientos básicos y aplicaciones - 56
(1)Configuración inicial de ROBOBASIC para ROBONOVA-I - 56
1)Configuración del puerto y elección del controlador - 56
2)Explicación de los puntos neutros y su configuración - 56
(2)Lista de comandos que aparecen en el programa plantilla - 56
1)Comando "goto AUTO" - 56
2)Tabla para las acciones y configuración de remocon - 57

V. Manual del software de ROBONOVA-I
1.Instalación y uso de ROBOBASIC V2.5 - 58
(1)Acerca de ROBOBASIC - 58
(2)Instalación de ROBOBASIC - 58
(3)Configuración inicial de ROBOBASIC - 60
(4)Programación con ROBOBASIC - 63
(5)Puntos neutros con ROBOBASIC - 64
(6)Control de servos en tiempo real - 68
(7)ROBOBASIC ROBONOVA-I Control de servos - 70
1)Control de los servos en tiempo real - 70
2)R0B0N0VA-I Control de servos - 72
(8)Control directo - 73

2.Uso de ROBOSCRIPT V2.5 - 74
(1)Acerca de ROBOSCRIPT - 74
(2)Configuración inicial de ROBOSCRIPT - 74
(3)Programación con ROBOSCRIPT - 76

3.Uso de ROBOREMOCON V2.5 - 78
(1)Acerca de ROBOREMOCON - 78
(2)Programación y uso de ROBOREMOCON - 78
(3)Uso de ROBOREMOCON en ROBOBASIC - 80

VI .Uso de programas - 82
1.Instalación y uso del programa plantilla - 82
(I)instalación de programa plantilla - 82
(2)Maneras de cambiar el programa plantilla - 82
2.Programación de aplicaciones - 86
(1) Captura de movimientos y su inclusión - 86
(2) Como IMPORTar un fichero ROBOSCRIPT - 89
(3) Conversión AD - 90
(4) Configuración de Gyros - 90
(5) Como usar I2C - 93
(6) Señales del protocolo HMI - 94
(7) Configuración de la alerta por voltaje bajo - 95

ROBOBASIC Manual de comandos
(Incluido en el CD)
Apartado 1 Sumario de comandos en ROBOBASIC - 3
Apartado 2 Sintaxis de ROBOBASIC - 10
Apartado 3 Explicación de los comandos de declaraciones ROBOBASIC - 21
Apartado 4 Explicación de los comandos de control de flujo - 25
Apartado 5 Explicación de las señales digitales I/O en ROBOBASIC - 44
Apartado 6 Explicación de comandos relacionados con la memoria - 55
Apartado 7 Explicación del módulo LCD en ROBOBASIC - 61
Apartado 8 Explicación de los comandos para el control de motores en ROBOBASIC - 72
Apartado 9 ROBOBASIC Comandos para la reproducción de música - 103
Apartado 10 ROBOBASIC Comandos para comunicaciones con el exterior - 114
Apartado 11 ROBOBASIC Comandos para el proceso de señales analógicas - 126
Apartado 12 ROBOBASIC Procesos, comandos y otros - 138
Apartado 13 ROBOBASIC Descripción de los comandos - 142

ROBONOVA-I

1. Fácil de montar
Su kit ROBONOVA-1 ha sido diseñado para montarlo fácilmente en 6 u 8 horas usando solo un destornillador.

2. Motores de los servos
Servos diseñados exclusivamente para el control de Robots
Este humanoide mecánico y totalmente articulado, de 12" de altura, se controla con 16 potentes servos digitales HSR-8498HB diseñados en exclusiva para el ROBONOVA-1 por Hitec. Estos servos personalizados incluyen un "set pin" para fijar de manera sencilla el brazo de los servos, protección contra tensiones y consumos elevados, piñones de carbonite y tecnología "feedback" para una sencilla programación.

3. Controladores, MR-C3024
Placa Micom con interface Flexible
La controladora es el corazón del ROBONOVA-1. Fijada en la parte trasera del robot, protegida por una fuerte placa de plástico, la controladora puede manejar hasta 24 servos y 16 módulos accesorios. Los dispositivos adicionales pueden ser gyros, sensores de aceleración, módulos de síntesis de voz y dispositivos funcionales como controladores Bluetooth y emisoras y receptores R/C. Este interface, flexible y sofisticado, permite al usuario convertir al ROBONOVA-1 en el robot de sus sueños. El firmware se puede actualizar conectándolo al PC.

4. Chasis de metal
Las pletinas de los servos, doradas y anodizadas, sirven hacen de exoesqueleto, ligero y resistente. ROBONOVA-1 también dispone de componentes de resistente plástico que protegen de daños a la batería y al controlador.

5. Batería y cargador
El ROBONOVA-1 se alimenta con una batería recargable de 5 celdas NiMH, respetuosa con el medio ambiente que permite una 1 hora aproximada de funcionamiento. El cargador de corriente continua incluido dispone de un circuito de protección para evitar problemas en la carga de la batería.

6. La clave - "Fácil de manejar"
ROBONOVA-1, La programación de Robots hecha fácil. Los usuarios de este excitante kit de Robot tienen la oportunidad de aprender robótica en primera persona. La programación es sencilla usando los programas RoboScript y RoboBasic mediante el cable de conexión al PC incluido.

1) RoboScript & RoboRemocon
Los que comienzan a programar robots estarán encantados de usar el software incluido de programación RoboScript. Sin conocer ningún lenguaje de programación, los usuarios podrán crear rutinas operativas con sólo el "clic de ratón". Use el programa RoboRemocon para controlar su ROBONOVA-1 con los datos operativos creados con RoboScript

2) RoboBasic
RoboBasic es una herramienta de programación basada en el lenguaje BASIC, y se suministra para usuarios avanzados. Contiene comandos específicos para el control básico del Robot. También puede usar RoboBasic junto a RoboScript para aumentar las prestaciones de la controlador suministrada Micom.

3) HMI (Hitec Multi-protocol Interface)
Actualización de firmware & varias configuraciones al conectarlo al motor del servo
Configuración de parámetros (2 tipos de ganancia, zona muerta, etc.) y cambio de valores durante el funcionamiento
Compatible con PWM(Pulse Width Modulation) utilizado en radio control
Control de un máximo de 128 servos conectados en cadena, Daisy chain serial interface, y PC oCapaz de retro-alimentarse con datos de voltaje, corriente y posición usando HMI

4) Función Catch & Play
Técnica para facilitar la programación
La manera más fácil de programar el ROBONOVA-1 es la función "catch and play". Usando RoboScript o RoboBasic, ponga el Robot en cualquier posición y pulse el ratón para "capturar" esa posición. Mueva el Robot a otra posición y repita el procedimiento. El software enlazará las distintas posiciones y generará un patrón de movimientos suaves para la transición de una a otra.

I. Antes de empezar

- Este manual contiene comentarios sobre el hardware del ROBONOVA-I y su montaje, sobre el controlador MR-C3024 y el software utilizado para controlar el Robot.
- Por favor, lea atentamente estas instrucciones y tenga cuidado al usar el producto. Este manual también se suministra en formato PDF dentro del CD, de manera que pueda copiarlo al disco duro o imprimirlo, siempre que lo necesite.
-Las características pueden ser modificadas sin previo aviso para realizar mejoras u otros motivos. Visite la página Web de Hitec para ver las últimas actualizaciones, (http://www.hitecrobotics.com)

Normas de seguridad

Las normas se dividen en dos categorías, avisos y precauciones, dependiendo del riesgo. Lea los avisos y mensajes de precaución con mucho interés antes de montar el kit.

⚠ Peligro: Se usa para indicar peligro mortal, lesiones graves, o daños a la propiedad, si no se respetan las instrucciones.

⚠ Advertencia: Se usa cuando puede haber lesiones leves o daños al equipo, si no se siguen las instrucciones.

⚠ Precaución: Se usa cuando puede haber lesiones leves o daños al equipo.

1. Notas sobre el montaje y uso

Este manual contiene información sobre prevención de lesiones y daños a la propieda

Peligro ⚠

- Respete las normas de seguridad cuando el Robot esté en funcionamiento.
- El buen funcionamiento del Robot no está garantizado, ya que se trata de un kit, que puede funcionar mal debido al montaje.
- Monte el kit en una zona con ventilación suficiente.

Advertencia ⚠

- Manténgalo lejos de los niños. Aunque parece un juguete, podría herir a un niño si no está bajo observación.
- Apague y desconecte la batería inmediatamente si hay problemas. Si el producto se rompe, se expone a la humedad o líquidos, llama, o calor, puede sufrir un shock.
- Nunca desmonte o modifique la batería y los cables. (Los cables son reparados por el servicio técnico.)
- Desconecte el cargador cuando no esté en uso.
- Nunca desmonte o manipule la circuitería de los servos.
- No lo use con mucho calor, humedad o frió. El kit está formado por componentes de precisión. En condiciones extremas puede producirse errores.
- Ponga mucha atención mientras construye el Robot. Ya que es un kit para montar, la seguridad y el funcionamiento no están garantizadas. Si está mal montado puede causar lesiones o daños.
- Compruebe que el conector de carga esté bien apretado. Quítelo en cuanto haya terminado el proceso de carga.
- Por favor, lea atentamente el manual. Compruebe la dirección de los servos y sus soportes. Si se montan al revés, desmóntelos y vuelva a montarlos.

Precaución ⚠

- Los servos incluidos en el kit requieren un mantenimiento regular para funcionar de manera óptima.
- Obtendrá lo mejor del Robot si lo usa en una superficie grande, plana y lisa. Si la superficie es pequeña o irregular, el Robot se podría caer y estropearse.
- Nunca sostenga el Robot durante su funcionamiento.
- No apague el controlador MR-C3024 mientras actualiza el firmware. El programa se estropearía (Contacte con el Servicio Técnico)

ROBONOVA-I

2. Seguridad al manejar baterías

- Este kit contiene una batería NiMH como fuente de alimentación. Las baterías NiMH son baterías recargables de alta potencia y necesitan cuidados especiales durante la carga y su almacenaje.

Carga:

- Conecte el cargador a una toma AC y conéctelo al controlador. La batería se carga a través del controlador. Si el cargador está conectado, el LED se pondrá rojo. Al terminar la carga, el LED se pondrá de color verde.
- Para cargar la batería fuera del controlador, necesitará un cable especial (Referencia 77102).

Advertencia ⚠

1) Manejo de las baterías

- La batería suministrada tiene una carga mínima. Una batería NiMh debe guardarse siempre con algo de carga.
- Si la batería NiMH se descarga por completo, y se almacena así un largo tiempo, su rendimiento disminuirá.
- Debe desconectar la batería del cargador y/o del controlador cuando no esté en uso. Guárdela batería en un sitio fresco y seco.

Advertencia ⚠

2) Tiempo de carga

- Se tarda unos 70 minutos en carga la batería por completo. Este tiempo puede variar dependiendo de la energía que quedase en la batería.
- Vigile las baterías durante la carga. Deténgala carga si la batería se calienta sospechosamente.
- Una sobrecarga puede estropear el paquete de baterías.

Precaución ⚠

3) Manejo de las baterías

- No modifique, ni desmonte el conector o cables de las baterías.
- Compruebe que ningún objeto extraño haga contacto entre los pines del conector, y que los cables no estén pelados.
- No exponga las baterías a temperaturas extremas o humedad. Guárdelas en un sitio fresco y seco.
- Manténgalas alejadas de otros elementos conductores, durante el transporte o almacenaje.
- Si las baterías se perforan o se deshacen, cámbielas.

Precaución ⚠

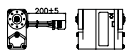
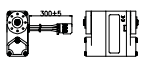
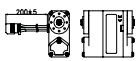
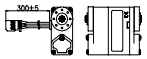
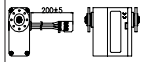
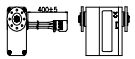
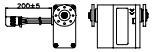
4) Primeros auxilios y reciclado

- Si las baterías se estropeasen y soltarán electrolito, entrando este en contacto con su piel u ojos, lávelos inmediatamente con abundante agua.
- Si entra en contacto con sus ojos, acuda al médico tan pronto como pueda.
- El electrolito de las baterías es tóxico. No solo daña al cuerpo humano, puede estropear el mobiliario.
- Si las baterías NiMh ya no pueden ser recargadas, deshágase de ellas consultando la normativa local.
- No las arroje en un incinerador.

ROBONOVA-I

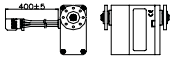
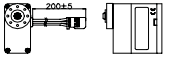
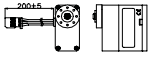
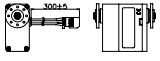
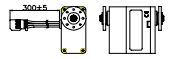
II. Notas sobre el montaje

ROBONOVA-I KIT Lista de partes

6 PCS	4 PCS	8 PCS	2 PCS	1 PCS	1 PCS	1 PCS
HR1B-0001	HR1B-0002	HR1B-0003	HR1B-0004	HR1B-0005	HR1B-0006	HR1B-0007
Tapped U Type Universal Bracket	Non-Tapped U Type Universal Bracket	I-Type Universal Bracket	H-Type Bracket	Back Body Frame	Front Body Frame	Top Body Frame
1 PCS	1 PCS	1 PCS	1 PCS	1 PCS	2 PCS	2 PCS
HR1C-0001	HR1C-0002	HR1C-0003	HR1C-0004	HR1C-0005	HR1C-0006 (Top Hand)	HR1C-0007 (Bottom Hand)
Front Body Cover	Back Body Cover	Goggle Cover	Front Head Cover	Back Head Cover	Top Hand Cover	Bottom Hand Cover
						
1 PCS	3 PCS	1 PCS	3 PCS	1PCS	1 PCS	1 PCS
HSR-8498HB1R200 (Sticker No.1)	HSR-8498HB1R300 (Sticker No.2)	HSR-8498HB1L200 (Sticker No.3)	HSR-8498HB1L300 (Sticker No.4)	HSR-8498HB2R200 (Sticker No.5)	HSR-8498HB2R400 (Sticker No.6)	HSR-8498HB2L200 (Sticker No.7)
130 PCS	12 PCS	6 PCS	2 PCS	40 PCS	12 PCS	4 PCS
PH/T-2 2*4 NI	PH/T-2 2*5NI	PH/T-2 2*8 NI	PH/T-2 2*26 BK	PH/M 2*4 NI	PH/M 2.6*4 NI	PH/M 3*4 NI
Pan Head Tapping Screw	Pan Head Tapping Screw	Pan Head Tapping Screw	Pan Head Tapping Screw	Pan Head Screw	Pan Head Screw	Pan Head Screw

*Los horns superiores ya vienen instalados en los servos HSR-8498HB

1 PCS	1 PCS
Pin Cover	Battery Wire Protector
For MR-C3024	

1 PCS	2 PCS	2 PCS	2 PCS	2 PCS	1 EA
HR1B-0008	HR1B-0009	HR1B-0010	HR1B-0011	HR1B-0012	MR-C3024
Bottom Body Frame	Shoulder Back Universal Bracket	Shoulder Front Universal Bracket	Hand Bracket	Foot Bracket	Controller
1 PCS	1 PCS	1 PACK	1 EA	1 EA	1 EA
HR1C-0008	HR1C-0009	Ni-Mh Battery (1,000mAh/ 6.0V/ 5 Cell)	CD-ROM	Interface Cable	Quick Charger
Right Foot Cover	Left Foot Cover	6.0 V 5Cell	ROBOBASIC & ROBONOVA-I User Manual	MR-C3024 Serial Interface Cable	6V/ 1,000mAh / 100~240V
					
1 PCS	1 PCS	1 PCS	1 PCS	1 PCS	2 PCS
HSR-8498HB2L400 (Sticker No.8)	HSR-8498HB3R200 (Sticker No.9)	HSR-8498HB3L200 (Sticker No.10)	HSR-8498HB2R300 (Sticker No.11)	HSR-8498HB2L300 (Sticker No.12)	Insert Bolt 3*4
					Insert Bolt
4 PCS	28 PCS	1 PCS	8 PCS	28 PCS	2 PCS
Support 3*5mm	FW 6*2.2*0.5 NI	HSR8498HA2	Cable Tie	Cable Clamp	Lug
Support	Flat Washer	Wheel Horn			

*Puede encontrar tornillos y arandelas adicionales en su ferretería

Diferencias entre los tornillos PH/M y PH/T.

PH/M (Rosca normal)

PH/T (Rosca chapa)

2. Notas sobre los servos y sus soportes (pletinas)

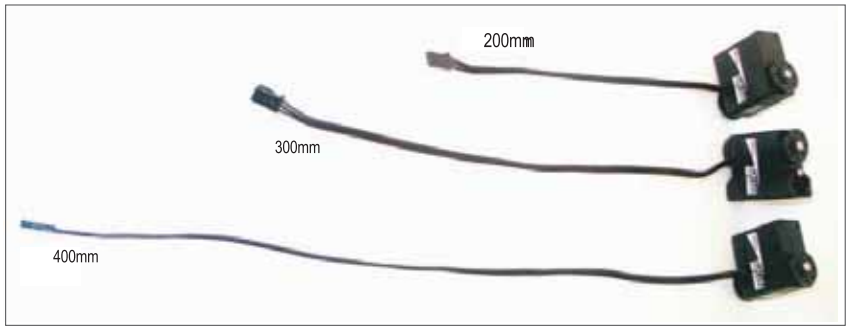
En los servos HSR-8498HB, la carcasa, el horn, la longitud los cables, y la dirección de éstos, están optimizados para el funcionamiento y montaje en cada articulación. Vea las fotos en el manual.

- El cable del servo está compuesto por 3 hilos, 2 negros y 1 gris. No quite el conector del cable.
- Durante el mensaje, mire las fotografías para comprobar la orientación del mensaje.

La lista siguiente detalla la configuración de los servos incluidos en el kit.

EL kit incluye 3 tipos diferentes de carcasas

3 cables con diferentes longitudes

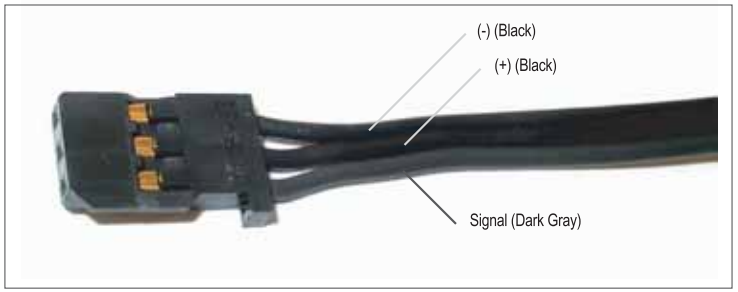


3 orientaciones distintas de los cables

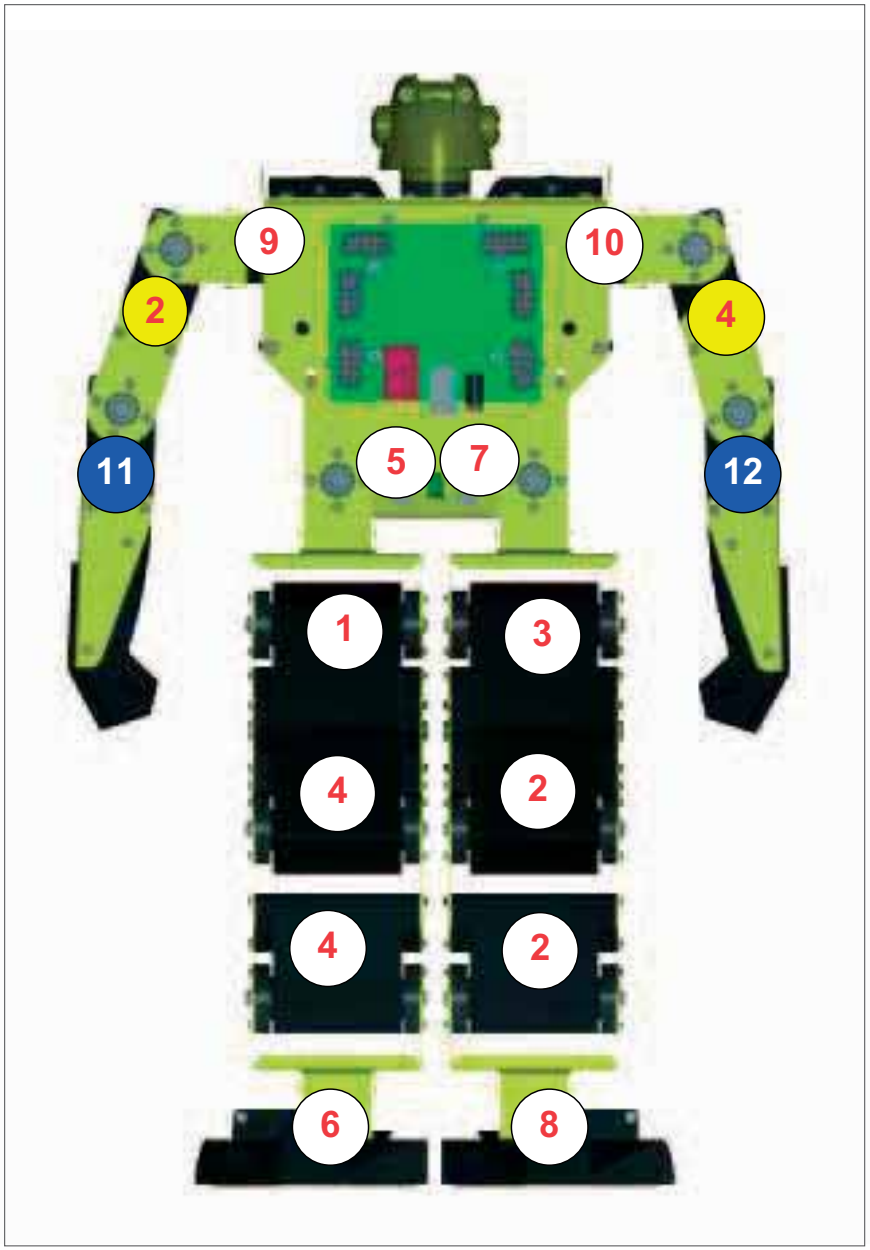
ROBONOVA-I

A la derecha no aparece "el símbolo CE" A la izquierda aparece "el símbolo CE"

El cable del servo está compuesto por 1 hilo gris oscuro (señal) y 2 dos hilos



Ubicación de los servos HSR-8498HB según el número de su pegatina
(Todos los servos HSR-8498HB están numerados con pegatinas según su carcasa, longitud y dirección de los cables para un mejor montaje.)



3. Herramientas necesarias para el montaje
Destornilladores Philips de alta calidad (PH #1 y PH #2), pinzas, alicates y fija tornillos.

Requisitos mínimos del PC
- Sistema operativo Windows
- AMD o Pentium 300MHz
- 60MB de espacio libre en el disco duro
- 128MB RAM

4. Especificaciones del HSR-8498MB

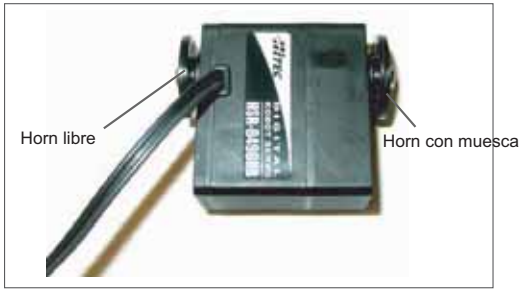
ESPECIFICACIONES
- Interface : Protocolo HMI, PWM
- Tensión de trabajo: 6.0V*
- Velocidad máxima: 0.20seg/60° a 6.0V
- Torque: 10kg.cm (138.87oz.in) a 6.0V
- Ángulo: Max 180°
- Peso: 55g (1.94oz)
- Dimensiones: 40 x 20 x 47mm
(1.57 x 0.78 x 1.85 in)

ESPECIFICACIONES DE PULSOS
- Pulso de control
  Neutral : 1500 µSeg , 0~180" :±1100~1900/µSeg
- Ciclo de pulsos: 12~26mSec (Normal : 21msec)

III. Montaje del kit

1. Ajuste de los horns de los servos
Existen cuatro tipos distintos de horns (dos con muescas distintas y dos libres) suministrados con los HSR-8498HB.

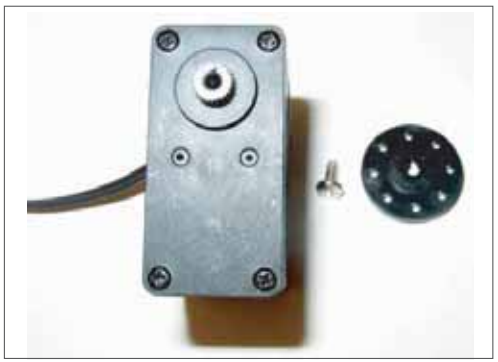
Hay número grabados sobre los horns con muesca.



Los números están situados cerca del tornillo.

Están codificados para un montaje sencillo y exacto.

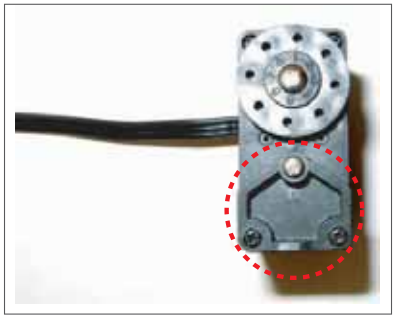
Una foto del servo con el horn con muesca quitado.
Se usa un tornillo BH/T 2.6x6mm para fijar el horn.



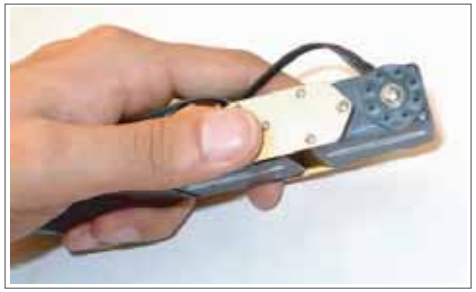
Los horns sin muesca giran libremente.

Una foto del horn libre suelto. Se usa un tornillo BH/T 2.6X6mm y una arandela (2.8x7.6mm) para fijar el horn.

El servo HSR-8498HB1 tiene una zona de montaje con forma pentagonal



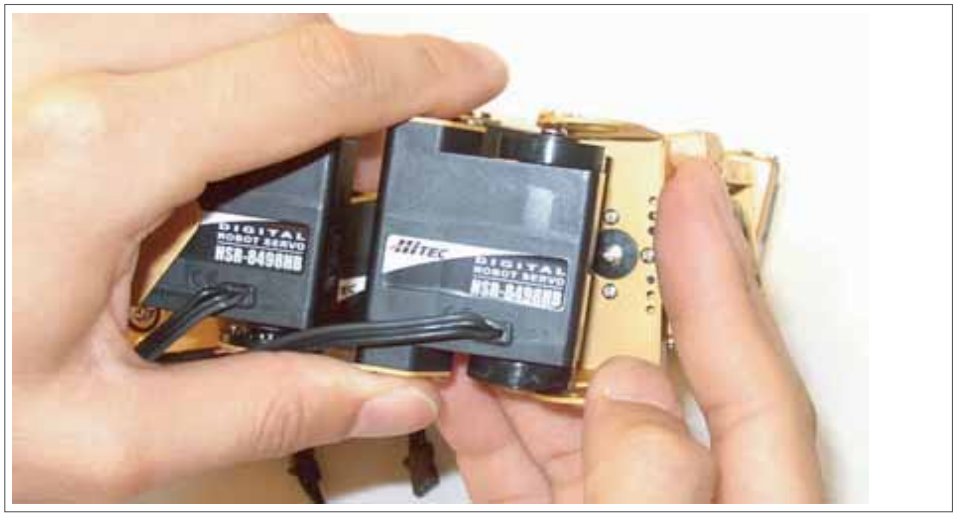
Para colocar una pletina, afloje los tres tornillos del pentágono y fije la pletina con ellos.



2. Colocar una pletina a un servo

(1) Coloque la pletina sobre los dos horns, con cuidado. Puede doblar suavemente la pletina para que se asiente sobre los horns.

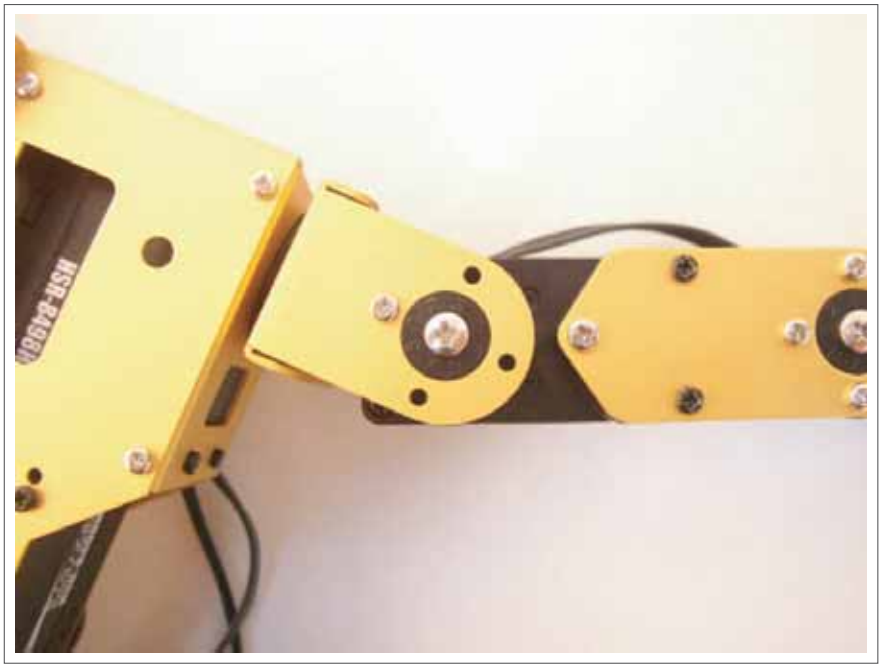
Cuando los tornillos se aprieten, volverá a recuperar su forma.

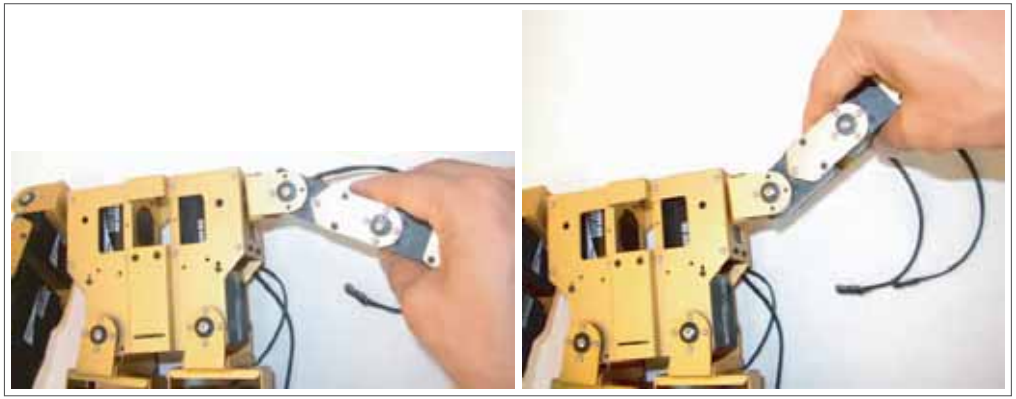


ROBONOVA-I

(2) Comprobación de los recorridos

Compruebe la longitud de los recorridos moviendo el servo manualmente.
Lea el apartado "comprobación de los recorridos" para confirmarlos en cada articulación.



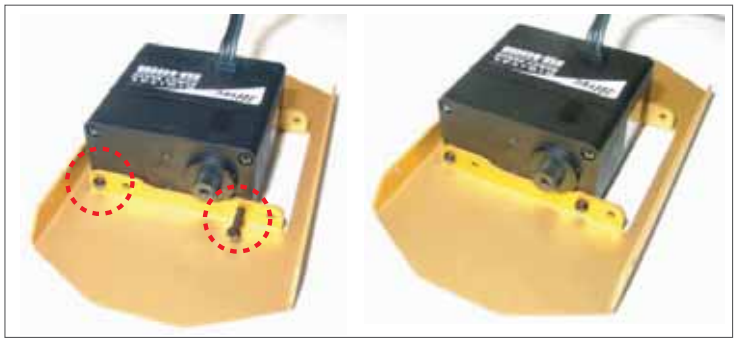


3. Montaje de las piernas.

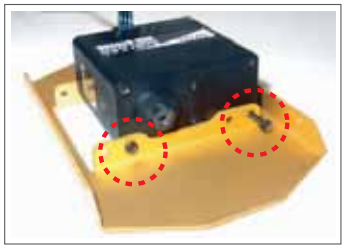
(1) Montaje de los pies derecho e izquierdo

Quite los dos tornillos negros del frontal, y los dos plateados de la trasera, de los servos 2L400 (Parte # 8, pié derecho) y 2R400 (Parte # 6,pié izquierdo) donde se engancharán las pletinas de los pies R1B-0012. Quite los tornillos de ambos horns de los servos. Quite el horn de los servos. Apunte donde van los tornillos negros y plateados, puesto que no son intercambiables.

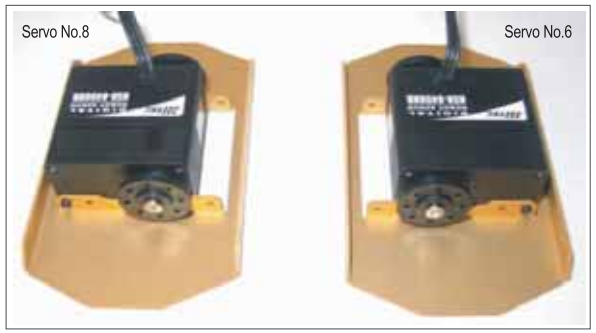
Ponga una pletina R1B-0012 en el servo 2L400 (Parte # 8) como se muestra en la foto, y vuelva a montar los cuatro tornillos a través de los agujeros de la pletina. Vuelva a colocar los horns en el servo. Al terminar, el cable del servo debe quedar por la parte superior.



Siga los pasos anteriores para montar el pié izquierdo.



Al terminar, los dos pies deben parecerse a los de la fotografía.

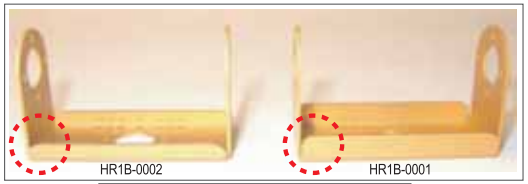


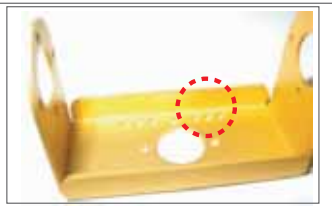
ROBONOVA-I

(2) Montaje del tobillo y del muslo. (4 juegos)

Para ver las diferencias entre los tipos de pletinas, fíjese en la foto. Las cuatro pletinas HR1B-0002 del lado izquierdo de la fotografía no tienen los agujeros tapados. Las seis pletinas HR1B-0001 tienen cuatro agujeros tapados cada una.

Otra manera de distinguir los dos tipos de pletinas, es comparar sus bordes laterales. Las HR1B-0001 tienen un borde mayor que las HR1B-0002.





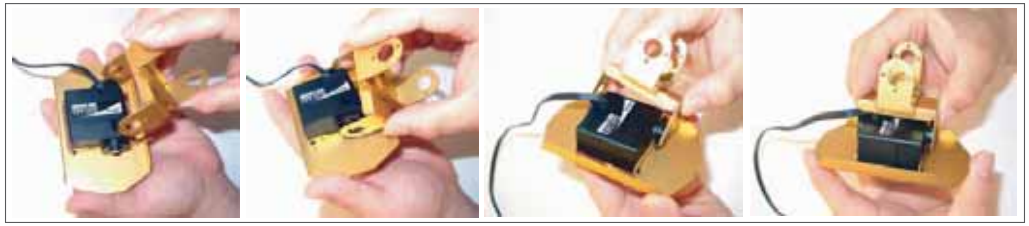
Hacen falta cuatro tornillos PH/M 2X4mm para unir una pletina HR1B-0001 con una 0002.

Monte las pletinas HR1B-0001 y 0002 como se ven la foto, usando cuatro tornillos PH/M 2X4mm para unirlas. Ponga mucha atención sobre donde vacada tornillo. Monte cuatro juegos.

(3) Montaje de los tobillos izquierdo y derecho.

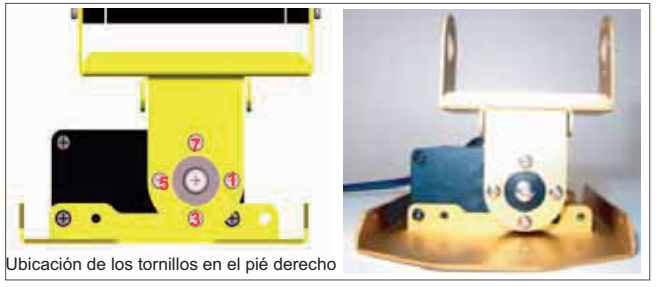
Montaje del tobillo derecho:
Monte los tobillos izquierdo y derecho del píe pre-montado. Use una pletina ya montada (tobillo) HR1B-0001 y atorníllela a los horns, frontal y trasero, de los servos 2L400 (pié derecho) y 2R400 (pié izquierdo). Serán necesarios ocho tornillos PH/T 2X4mm.

Use la foto para guiarse, Sitúe las pletinas del tobillo sobre los horns de los servos con cuidado. Si hiciese falta, puede abrir un poco la pletina para que encaje en los horns. Cuando la fije a los horns con los tornillos, volverá a tomar su forma.



Compruebe la orientación del cable del servo y la forma de la pletina. Guíese por la foto.

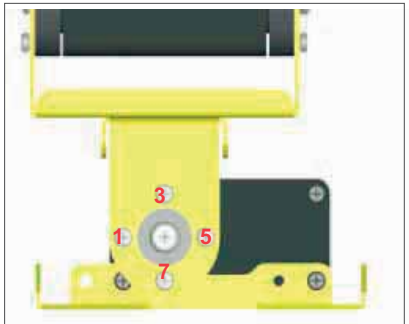
Fije la pletina al horn del servo frontal usando cuatro tornillos PH/T 2X4mm según la foto. Preste atención a la posición de la pletina y del horn del servo frontal. El punto # 1 del horn debe estar a las 3 en punto para el pié derecho y a las 9 en punto para el izquierdo. Consulte las fotos.



Ubicación de los tornillos en el pié derecho

Use otros cuatro tornillos PH/t 2x4mm para fijar la pletina al horn libre.

Al terminar el montaje, los pies y los tobillos deben quedar como en la fotografía. Una vez montados, los cables de cada servo deberían ir paralelos a la planta de cada pié.

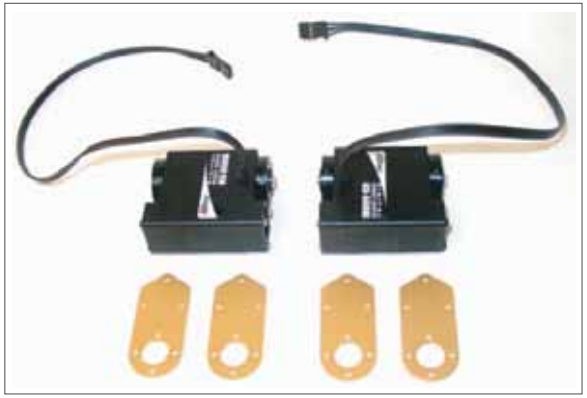


Ubicaciones de los tornillos para el pié izquierdo

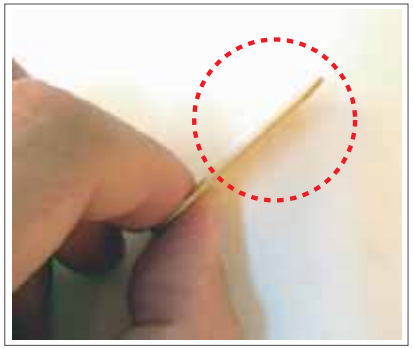
Montaje del tobillo izquierdo

(4) Montaje de la tibia

Componentes necesarios para el montaje de la tibia: 4 pletinas HR1B-0003 y un servo 1L300 (Parte #4, tibia izquierda) y otro 1R300 (Parte #2, tibia derecha).



La pletina HR1B-0003 tiene un lado con los bordes pronunciados y otro con bordes suaves. Por su seguridad, el lado con bordes suave de la pletina debe apuntar hacia el lado externo del Robot.



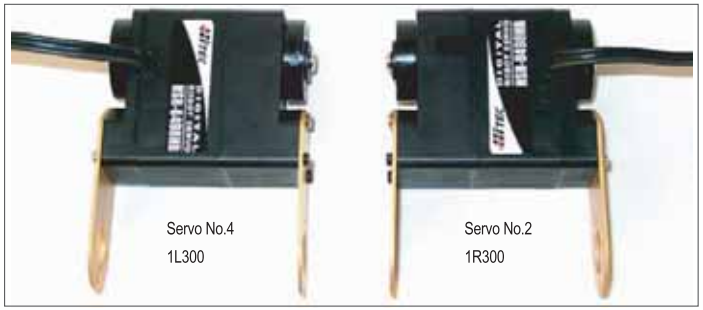
Quite los seis tornillos fijados a la parte pentagonal del servo 1L300. Hay tres tornillos plateados en la parte trasera, y uno plateado y dos negros en el frontal. Monte las dos pletinas HR1B-0003, una por cada lado de los servos y vuelva a colocar los tornillos.

Use las fotografías para guiarse sobre los puntos donde colocar los tornillos. Fíjese en la posición de los tornillos plateados y negros.

Frontal del servo

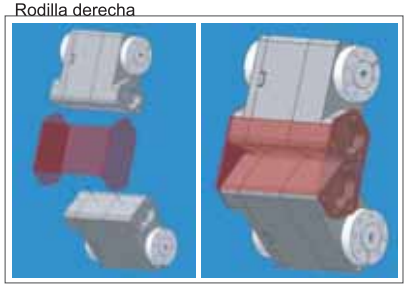
Trasera del servo

Monte la ora tibia usando el servo 1R300. Ambas tibias se montan de la misma manera.

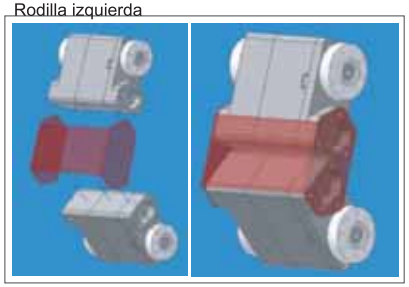


(5) Montaje de la rodilla

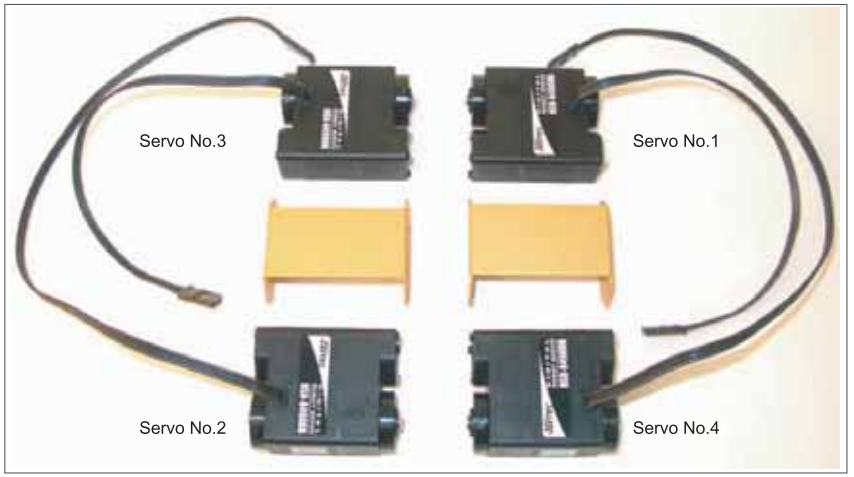
Rodilla derecha



Rodilla izquierda

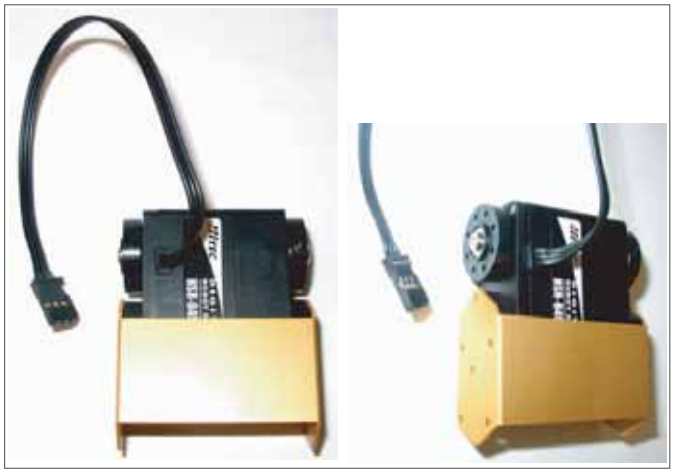


Componentes necesarios para el montaje: 1-1L200 (Parte #3, servo superior de la rodilla derecha), 1-1R300 (Parte #2, servo inferior de la rodilla derecha), 1- 1R200 (Parte #1, servo superior de la rodilla izquierda), 1- 1L300 (Parte #4, servo inferior de la rodilla izquierda) y 2 pletinas HR1B-0004 para las rodillas.



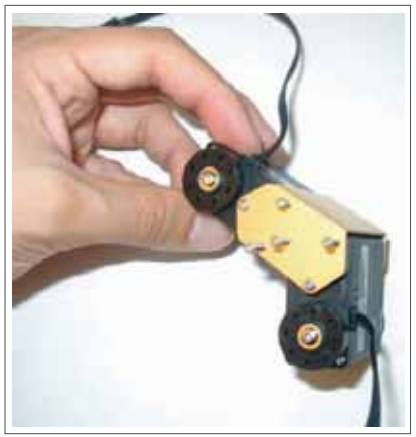
Quite los seis tornillos de los pentágonos, frontal y trasero, del servo para preparar el montaje de la rodilla.

Según la foto, debe unir una pletina para la rodilla HR1B-0004 a los servos No. 3 y No. 1 y fíjela usando los tornillos que había quitado anteriormente, compruebe la ubicación de los tornillos plateados y negros.

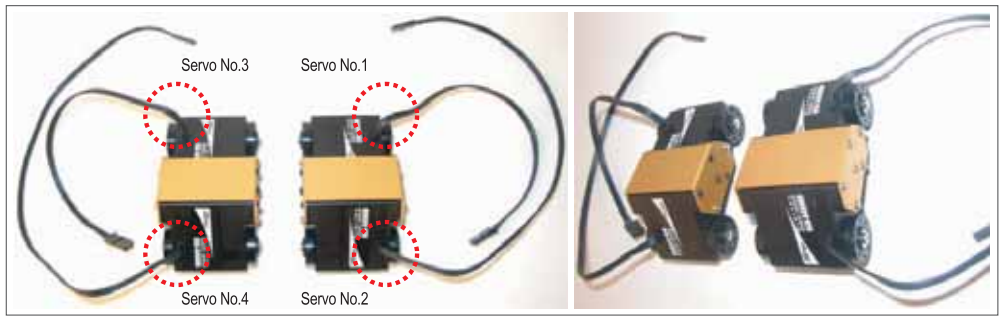


Quite los tornillos de los pentágonos, frontal y trasero, del servo 1R300 (Parte #2, rodilla inferior derecha) y del 1L300 (parte #4, rodilla inferior izquierda) . Después de revisar cuidadosamente las fotografías una los servos 1R300 y 1L300 a la pletina HR1B-0004 asegurándose que los servos están por debajo de la rodilla con la orientación correcta

Fíjelos con los tornillos quitados anteriormente y fijándose en los colores y posición de los tornillos

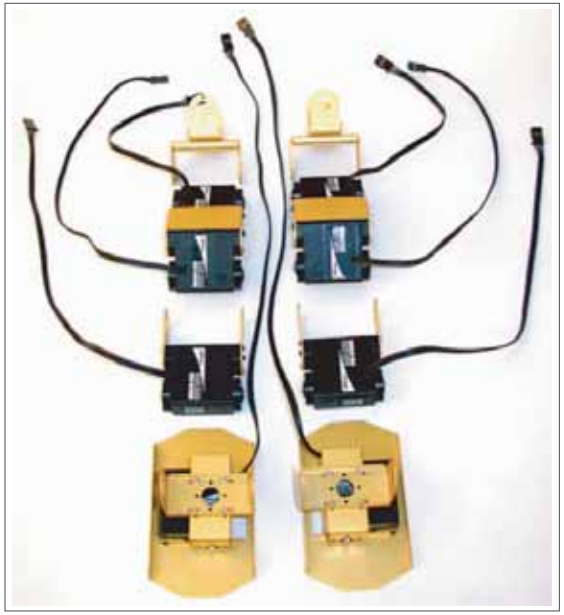


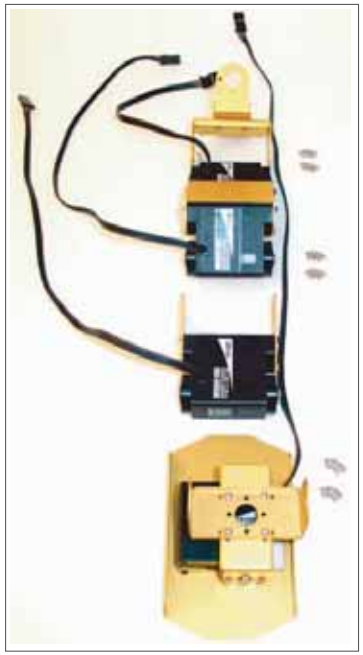
Compruebe que la dirección de los cables de cada servo coincide con los de la fotografías.



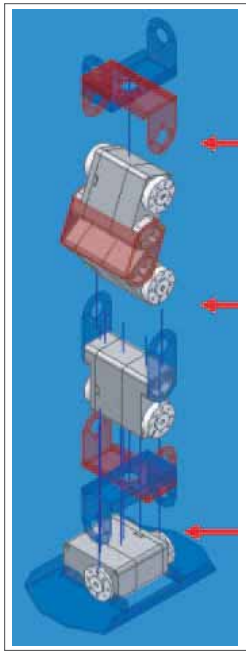
(6) Montaje de la pierna completa.

Monte primero la pierna derecha. Coloque las partes como en la fotografía, antes de comenzar con el montaje.





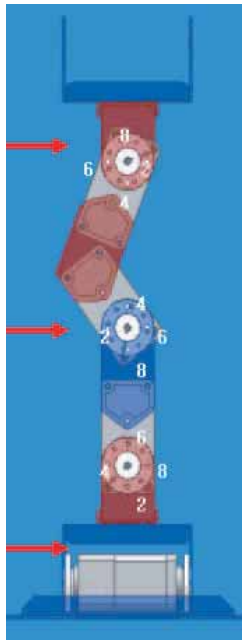
Ponga las pletinas y los horns frontales de los servos según los número grabados en los horns de los servos.



Conecte las rodillas a las pletinas de los muslos

Conecte la tibia a las rodillas

Conecte el tobillo a la tibia



Unión del pié y la tibia

- Conecte el horn frontal del servo del servo de la tibia derecha (1R300, #2) con la pletina del pié derecho.
- Fíjese en la foto para comprobar la orientación. Fije la pletina al horn usando cuatro tornillos PH/T 2X4mm.
- Fije la pletina al horn libre del servo con cuatro tornillos PH/T 2X4mm.

Unión de la tibia y la rodilla

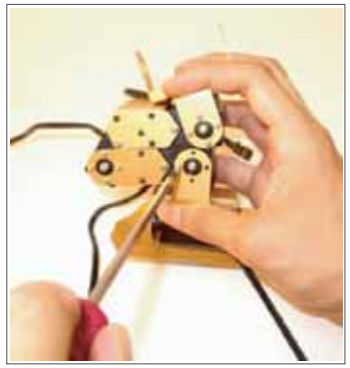
- Una la pletina de la tibia HR1B-0003 a los horns del servo (1R300, #2) de la parte inferior de la rodilla.
- Fíjese en la foto para comprobar la orientación. Fije la pletina al horn usando cuatro tornillos PH/T 2X4mm.
- Fije la pletina al horn libre del servo con cuatro tornillos PH/T 2X4mm

Unión de la rodilla y el muslo

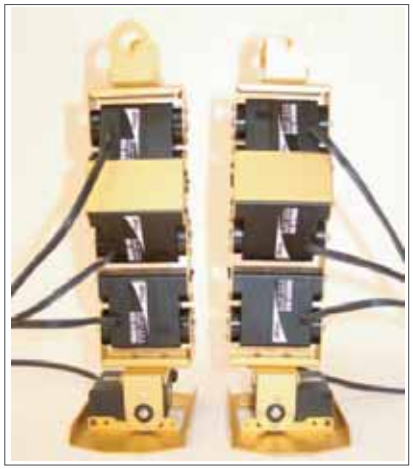
- Conecte las pletinas del muslo, que había montado anteriormente, a los horns del servo (1L200, #3) superior de la rodilla.
- Fíjese en la foto para comprobar la orientación. Fije la pletina al horn usando cuatro tornillos PH/T 2X4mm.
- Fije la pletina al horn libre del servo con cuatro tornillos PH/T 2X4mm

¡Cuidado! Las piernas solo quedarán bien montadas si el recorrido de los servos es completo.

Si el Robot no puede mover las rodillas completamente tal y como aparece en la fotografía, suelte los tornillos de las pletinas y ajuste la posición de los horns de los servos. Una vez hecho, vuelva a fijar las pletinas con los tornillos.

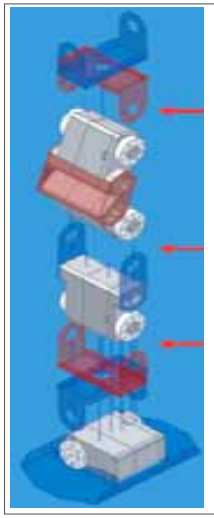


Vista frontal de las piernas montadas.



Ponga las pletinas y los horns frontales de los servos según los números grabados en los horns.

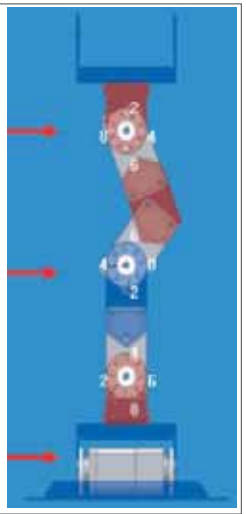
La pierna izquierda se monta exactamente igual que la derecha. Anote la posición de los números grabados en el servo durante el montaje.



Conecte las rodillas a las pletinas de los muslos

Conecte la tibia a las rodillas

Conecte el tobillo a la tibia



(7) Cableado y montaje de la tapa de los pies.

En este paso se ordenan los cables y se instalan las tapas de los pies del ROBONOVA-I.

Prepare dos sujeciones para los cables (cintas transparentes) y dos arandelas 2.2x0.5mm.

Monte primero el pié derecho. Quite el tornillo que está por debajo del horn libre.

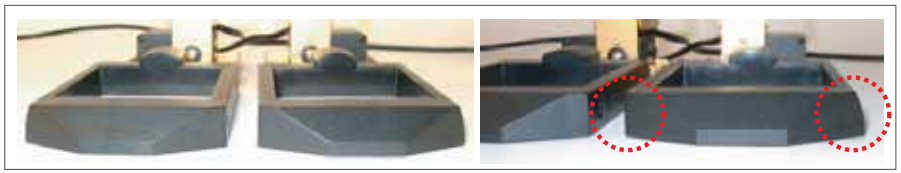
Doble la sujeción sobre el cable.

Coloque el tornillo que quitó, con una arandela, a través de los agujeros de la sujeción de los cables y fíjelo al horn libre del servo.

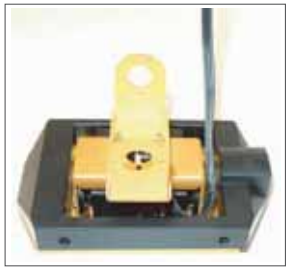
Aspecto de los pies con las sujeciones de los cables instaladas.

Puede colocar ahora las tapas de los pies HR1C-0008(lado derecho), 0009(lado izquierdo) usando cuatro tornillos PH/M 2X4mm para cada una.

Ponga las tapas sobre los pies. El lado plano de la tapa debe mirar con el lado biselado hacia fuera.



Compruebe la dirección del cable de cada pié. No debe estar en contacto con las tapas.



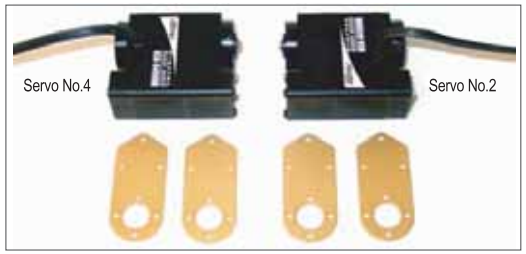
Fije la tapa al pié con 4 tornillos 4 PH/M 2x4mm.

Pié terminado.

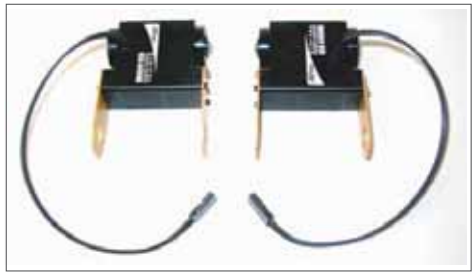
4. Montaje de los brazos

(1) Montaje de los hombros
Quite los seis tornillos de los pentágonos de los servos 1R300

(Parte #2, lado izquierdo) y 1L300
(Parte #4, lado derecho) y fije las pletinas HR1B-0003 con los tornillos que ha quitado. Fíjese en la ubicación de los tornillos plateados y los negros.



Monte dos juegos. Los hombros derechos e izquierdo se montan de la misma manera.



(2) Montaje del antebrazo
Quite los cuatro tornillos de los dos servos 2R300 (Parte #11) y 2R300 (Parte #12). Los tornillos están en la parte inferior de los servos. No quite los tornillos que están junto al horn. Coloque las pletinas HR1B-0011 y fíjelas con los tornillos que ha quitado. Fíjese en la ubicación de los tornillos plateados y negros al montarlos de nuevo. Compruebe que los cables de ambos servos van hacia el interior del Robot.

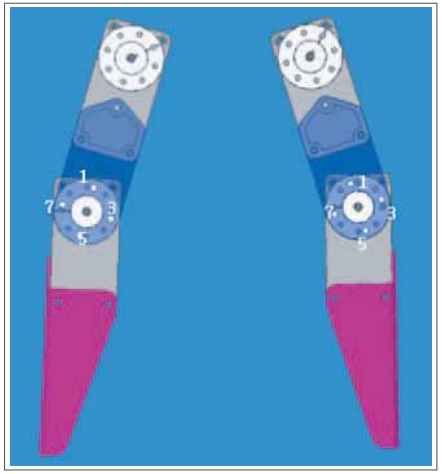
Para montar el antebrazo derecho, apriete las piezas HR1C0006 y 0007 entre sí, y fíjelas a la pletina 0011 con cuatro tornillos PH/T 2X5mm.

El antebrazo izquierdo se monta de la misma manera. HR1C- 0006 y 0007 se usan para el lado izquierdo.

(3) Montaje del brazo completo

Conecte el hombro, montado anteriormente, y los antebrazos usando 8 tornillos PH/T 2X4mm por brazo.

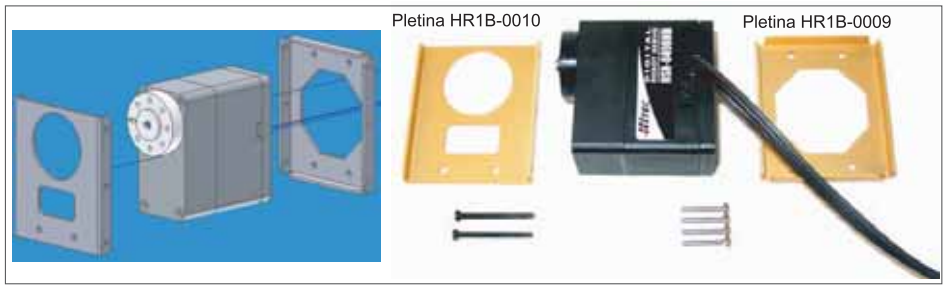
Los cables de los servos, de los hombros y antebrazos, deben mirar hacia el exterior. Use la foto como referencia, ajuste los horns frontales de los servos, según los números grabados en estos para que coincidan con los del gráfico y fíjelos con cuatro tornillos PH/T 2X4mm a la pletina. Fije los horns libres del servo usando cuatro tornillos PH/T 2x4mm.



Cuando estén montados, los brazos deben parecerse a los de la foto.

5. Montaje del cuerpo

(1) Unión del hombro con el interior del cuerpo
Quite los pernos de los servos 3R200(No.9) y 3L200(No.10). A continuación, monte las pletinas HR1B0009 y 0010 en el servo.

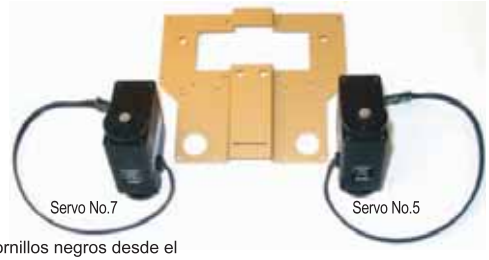


Usando la foto como referencia, quite los dos tornillos negros del frontal de los servos 3R200(No.9) y 3L200(No.10).
No quite los tornillos situados cerca del horn. Fije la pletina HR1B-0010 al servo, volviendo a colocar los tornillos.
Para instalar las pletinas HR1B-0009 en los servos, tiene que quitar los cuatro tornillos plateados de la parte trasera de cada servo, y volverlos a usar para fijar la pletina.
Monte dos.

ROBONOVA-I

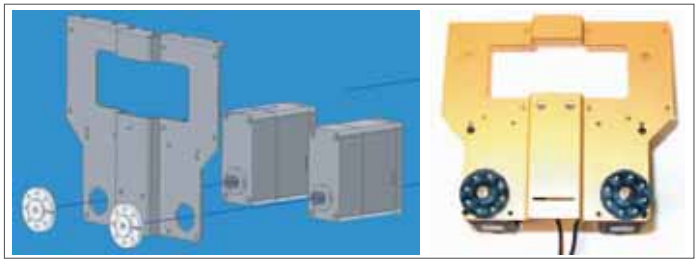
(2) Montaje de la cadera frontal.

Prepare los servos 2R200 (Parte #5, cadera izquierda) y los 2L200 (Parte #7, cadera derecha) para su montaje.

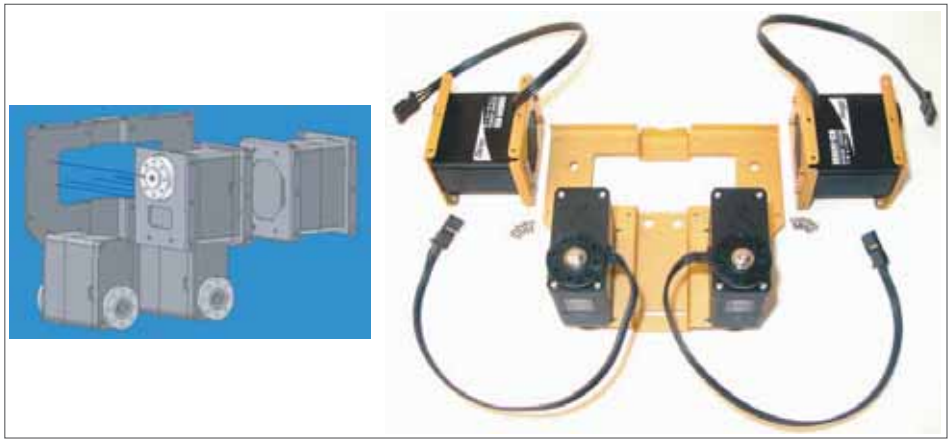


Quite los horns frontales de cada servo y quite los dos tornillos negros desde el borde frontal.

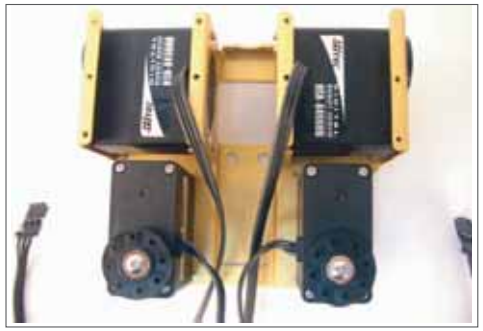
Monte los servos en la pletina del cuerpo HR1B-0006 reinstalando los tornillos y los horns. Los cables de los servos deben apuntar hacia el interior del cuerpo.



Fije los hombros a la pletina frontal del cuerpo usando cuatro tornillos PH/M 2X4mm para cada uno.



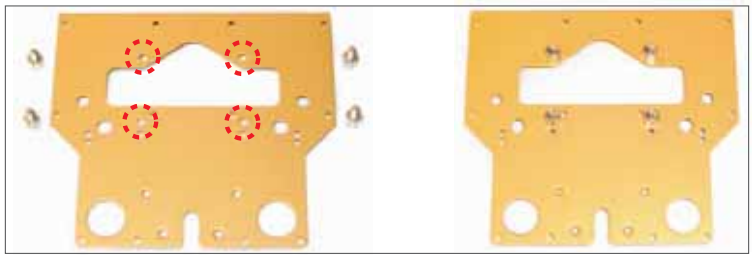
Fíjese en la dirección de los cables de los servos. Ambos debe ir hacia el interior del cuerpo.



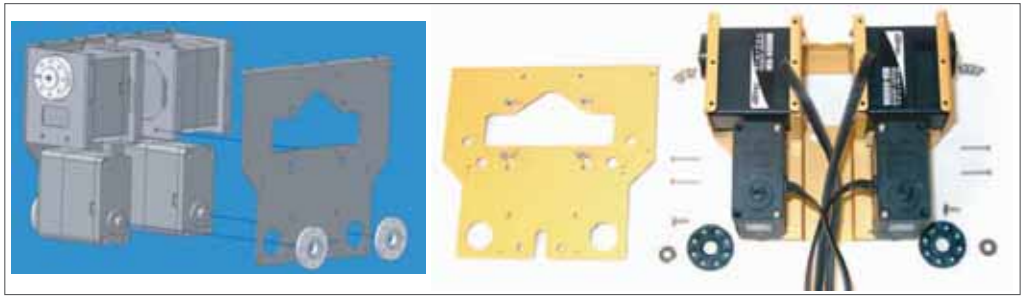
Aspecto una vez montado.

(3) Montaje de la parte trasera de la cadera

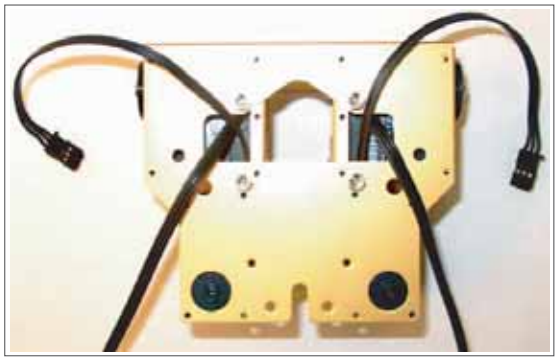
Fije los soportes de 5mm-3Ø en la pletina trasera del cuerpo HR1B0005 que sostendrán el controlador MR-C3024.



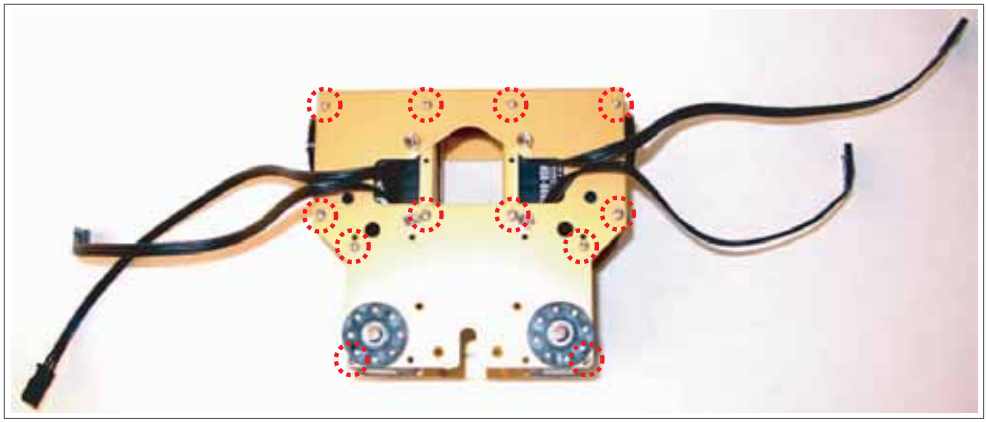
Quite los horns libres de los servos 2R200 (No. 5) y 2L200 (No. 7). Quite los dos tornillos situados en el borde del servo.



Usando la imagen como referencia, ponga la pletina trasera del cuerpo HR1B-0005 sobre el cuerpo.

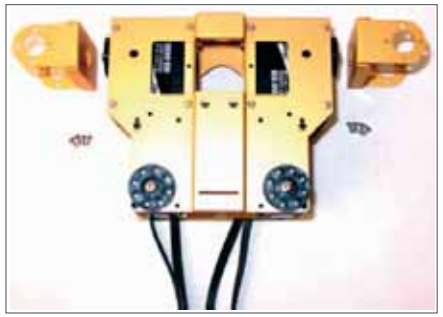


Use ocho pernos PH/M 2X4mm para fijar la pletina trasera del cuerpo HR1B0005 a los hombros y reinstale los tornillos de la cadera que quitó anteriormente. Coloque de nuevo los horns libres.

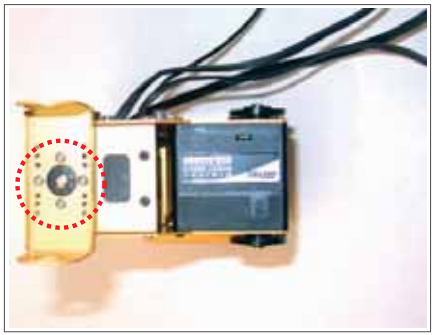


(4) Montaje de los hombros.

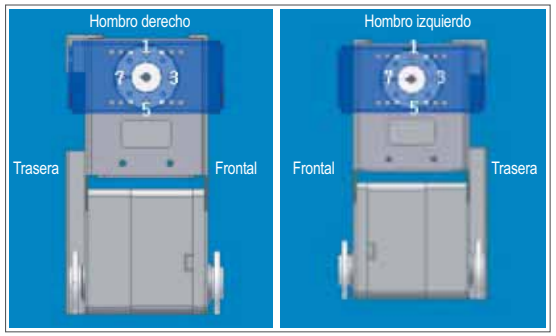
Monte una pletina HR1B-0002 en cada servo 3L200 (Parte#10, hombro derecho) y R200 (Parte#9, hombro izquierdo) instalados en el cuerpo



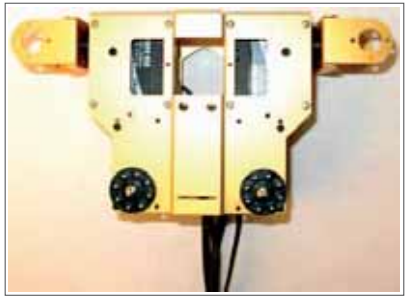
Usando la foto como referencia, fije las pletinas a los horns de los servos usando cuatro tornillos PH/T 2X4mm.



- Este gráfico muestra la posición de los horns en las pletinas.

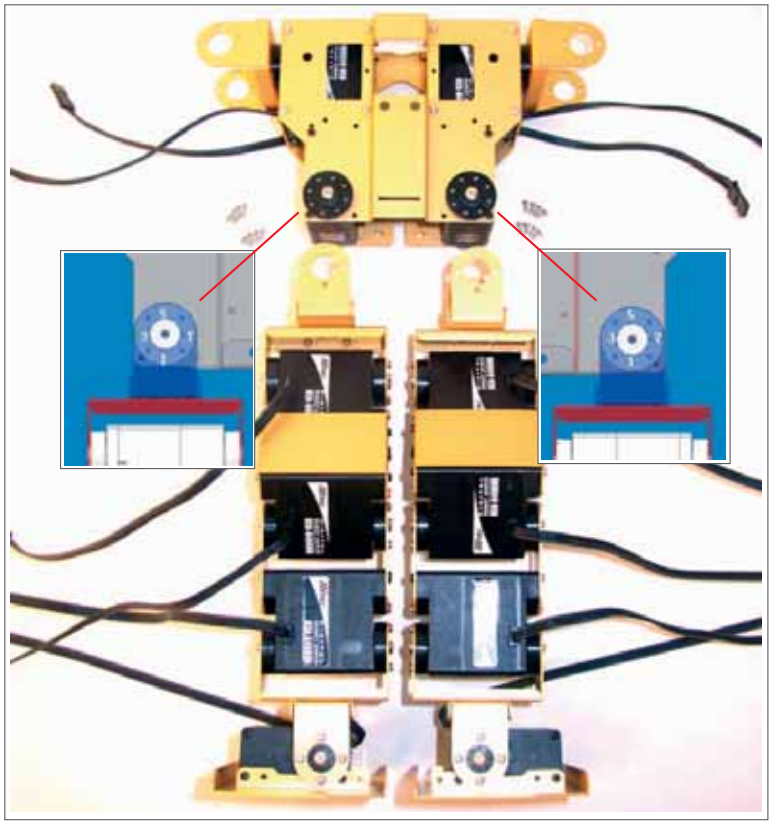


Aspecto del cuerpo una vez montado.

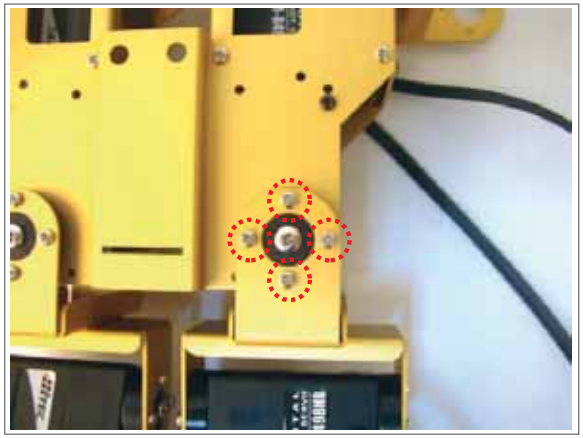


6. Uniendo los brazos y las piernas con el cuerpo

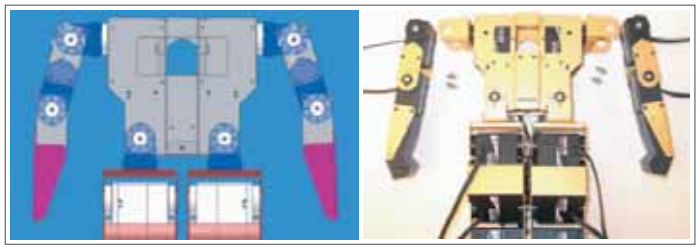
(1) Unión de piernas y cuerpo



Compruebe en la imagen la correcta posición de los horns al unir las piernas con el cuerpo. Fije las pletinas a cada horn de los servos (libres y con muesca) usando cuatro pernos PH/T 2X4mm.



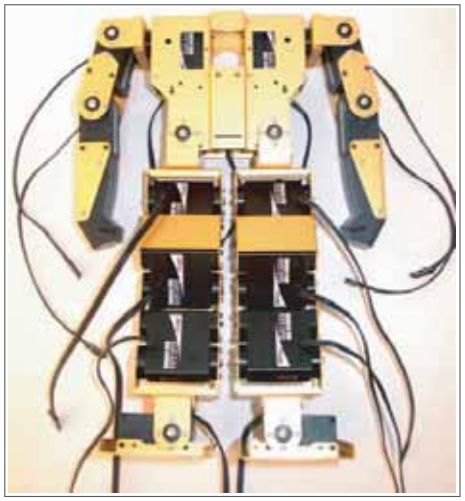
(2) Unión de brazos y cuerpo



Compruebe en las fotos la posición de los horns cuando una los brazos al cuerpo del Robot. Fije las pletinas a los horns (con ranura y libres) con cuatro pernos PH/T 2X4mm.

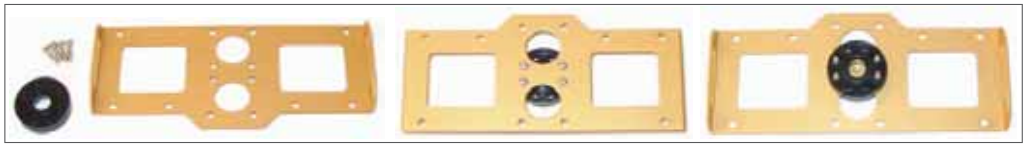
Después de fijar los servos a las pletinas, compruebe el recorrido de los servos manualmente.
Los brazos estarán bien montados cuando puedan moverse 180 grados.

Aspecto una vez terminado



7. Unión de la cabeza y el cuerpo

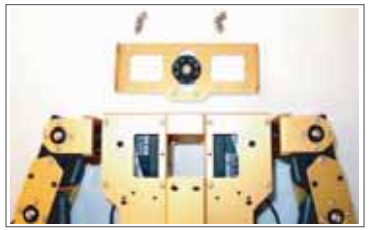
Conecte el horn del servo HSR8498HA2 a la parte superior del cuerpo HR1B-0007 usando cuatro tornillos PH/T 2X8mm.



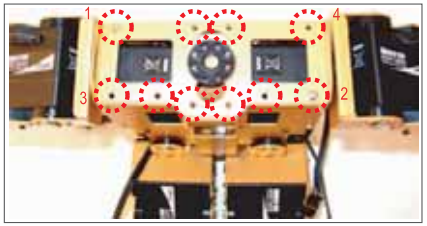
Conecte la placa de LEDs al visor (parte # HR1C-0003) con dos tornillos PH/T 2X4mm.

Encaje la visera en la cabeza, presionando despacio desde el frontal.

Fije la pletina HR1B-0007 al cuerpo usando diez pernos PH/T 2.6X4mm.



Para colocar adecuadamente la pieza HR1B-0007, comience colocando los tornillos en forma de cruz, comenzando en una esquina y siguiendo por la diagonal. Instale los demás tornillos.



Fije la parte frontal de la cabeza con dos tornillos PH/T 2X8mm. No apriete demasiado.

Fije la parte trasera de la cabeza con la parte frontal usando dos tornillos PH/T 2X5mm.

Aspecto del Robot

8. Colocando la tapa frontal del cuerpo.

Esta tapa protege el cuerpo

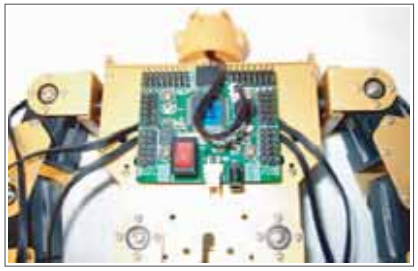
La parte frontal HR1C-0001 se une al cuerpo usando dos tornillos PH/T 2X4mm. Esta imagen muestra la posición de los agujeros en la tapa para los tornillos.

Usando un destornillador Philips, delgado y largo, apriete los dos tornillos PH/T 2x4mm desde la espalda del Robot en la tapa delantera.

Tapa frontal del cuerpo instalada.

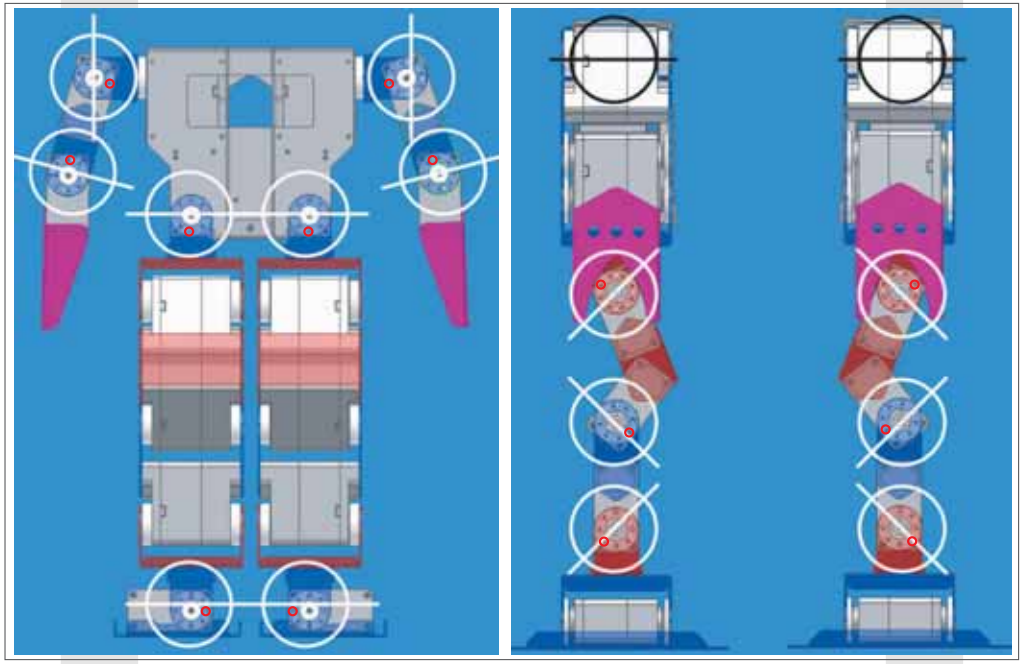
9. Instalación del controlador en el Robot.

Fije el controlador MR-C3024 en la espalda del robot usando cuatro pernos PH/M 3X4mm.



Comprobación del recorrido de los servos

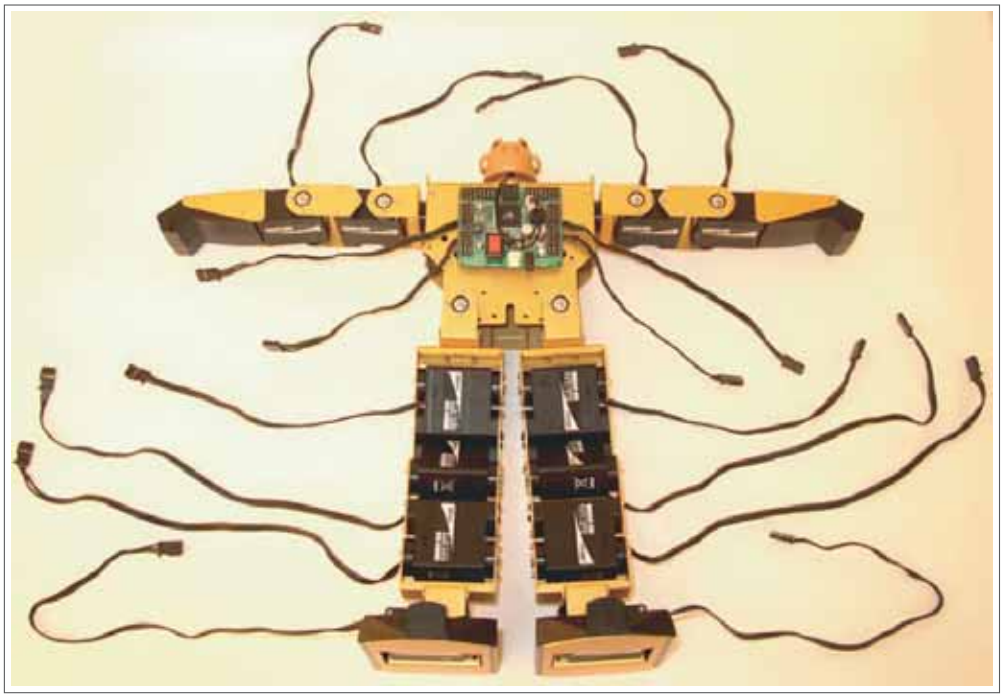
* En las imágenes, los servos se mueven 180 grados.
* Mueva, a mano, cada articulación para comprobar los recorridos.



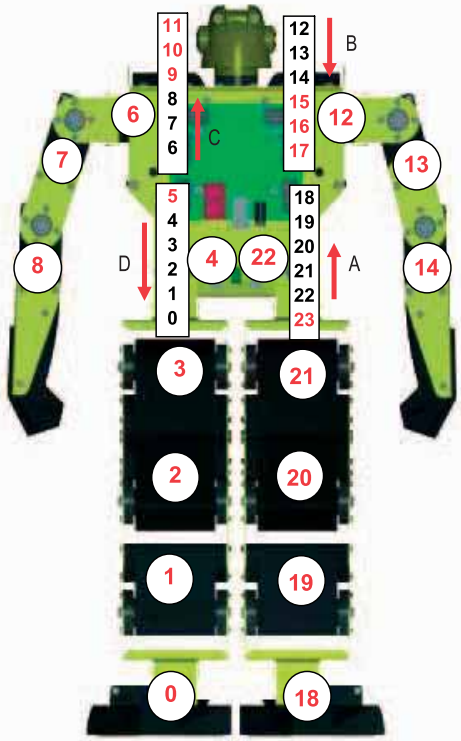
10. Cableado

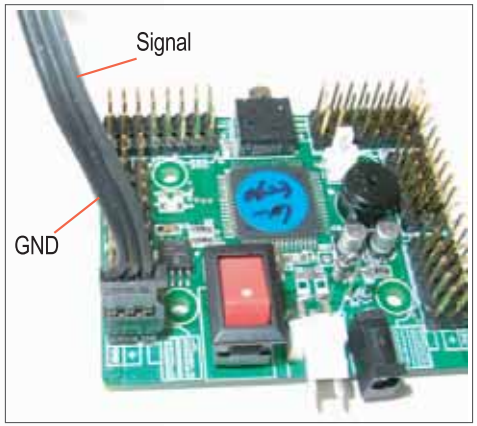
El RoboNova-1 tiene 16 cables que se conectan al controlador.
Es importante para el funcionamiento del Robot que esos cables queden bien sujetos al cuerpo.
Conexiones al puerto del MR-C3024 y posiciones de las presillas y sujeciones de los cables.

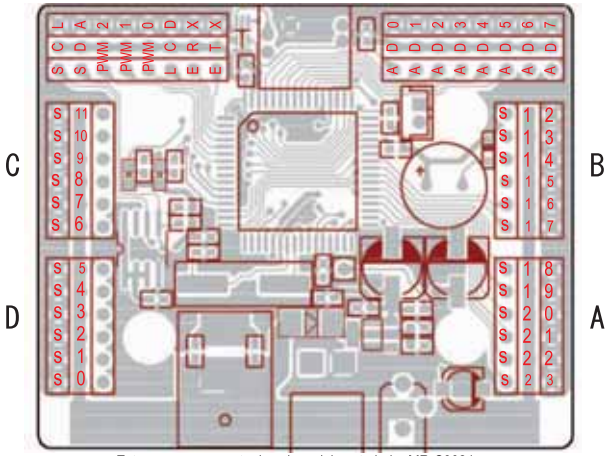
(1) Distribución de los cables.
Prepare los cables para su conexión al controlador



Fíjese en el diagrama de conexiones para saber en que conector deben ir los cables.
Los cables de color gris (hilos) llevan las señales.

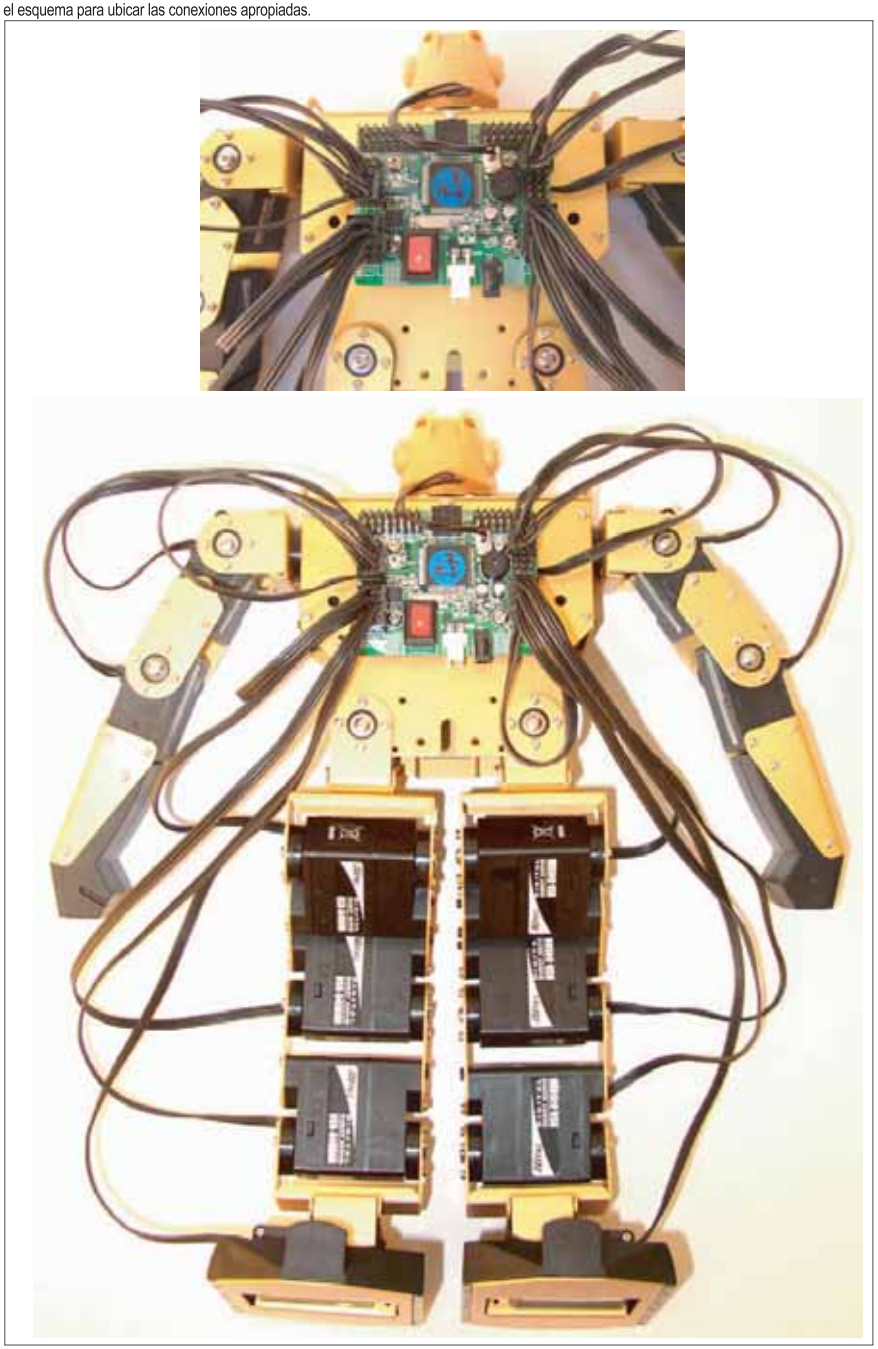






Este esquema muestra los pines del controlador MR-C3024.

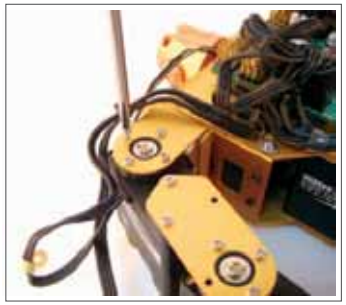
Conecte los servos en los puntos apropiados del controlador.
Consulte el esquema para ubicar las conexiones apropiadas.



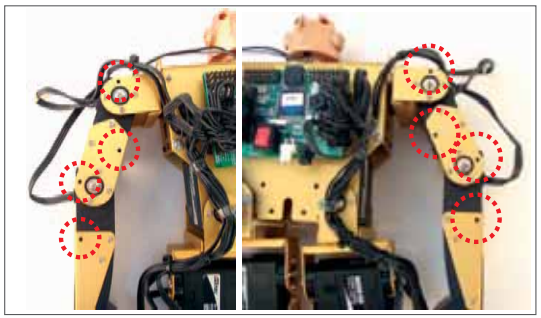
2) Instalación de las sujeciones de los cables.

La instalación de estas sujeciones evitarán que los cables interfieran con los movimientos del Robot. Vea el apartado 1 para ver como se colocan las sujeciones.

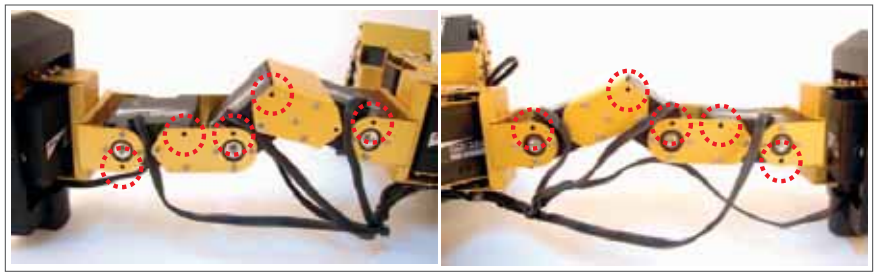
Usando las fotos como referencia, quite los tornillos para fijar las sujeciones de los cables.



Esta imagen muestra las ubicaciones de las sujeciones de los cables en ambos brazos.



Esta imagen muestra las ubicaciones de las sujeciones de los cables en ambas piernas.

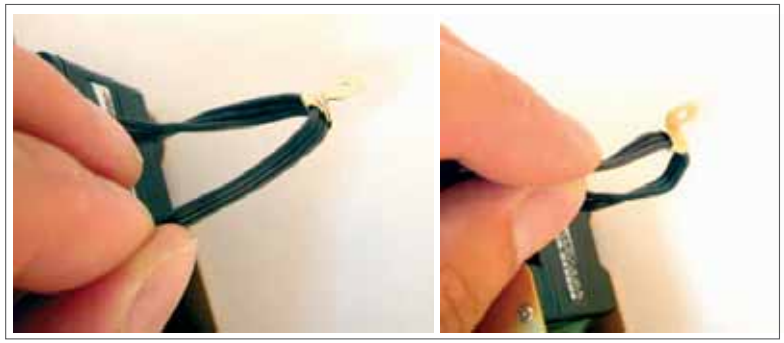


Piezas necesarias para fijar las sujeciones a un horn del servo.

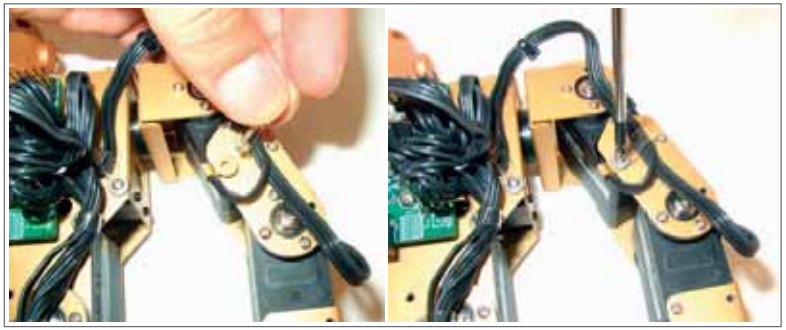
Instale una sujeción de los cables sobre el cuerpo, en los lugares marcados en las fotografías.

Para los servos No.2 y No.4 de los brazos, use una orejuela (pieza dorada brillante) y una sujeción.

Enrolle primero la orejuela sobre el cable.



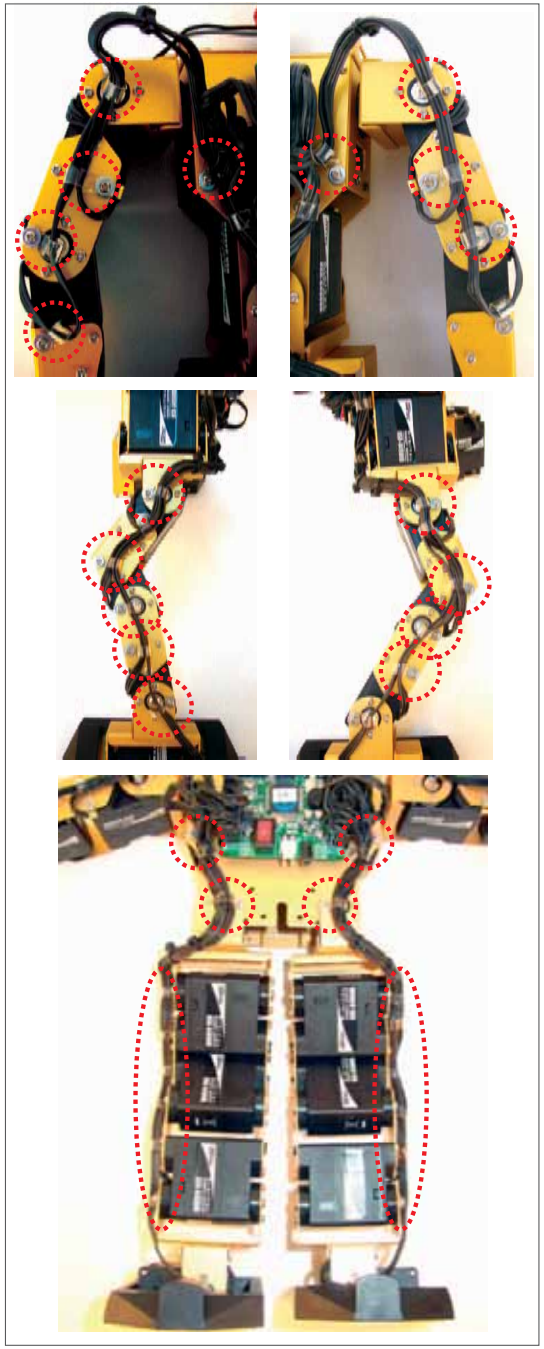
Use la foto como ejemplo de como unir la orejuela y la sujeción de cables.



Orejuela y cables montados.

Fije las sujeciones a cada uno de los horns de la cadera, como se aprecia en la foto.

Montaje de la sujeción finalizado.



En primer lugar, tire un poco de los cables que vienen de los hombros y cójalos con una presilla.
Suelte los pernos del controlador MR-C3024 y lleve 3~4 de los cables más largos por debajo del MR-C3024 ,volviendo a apretar los pernos.

Coloque el resto de los cables alrededor del controlador como en la foto. Fije los cables que unen las piernas al cuerpo con una presilla para disminuir el roce con las pletinas.

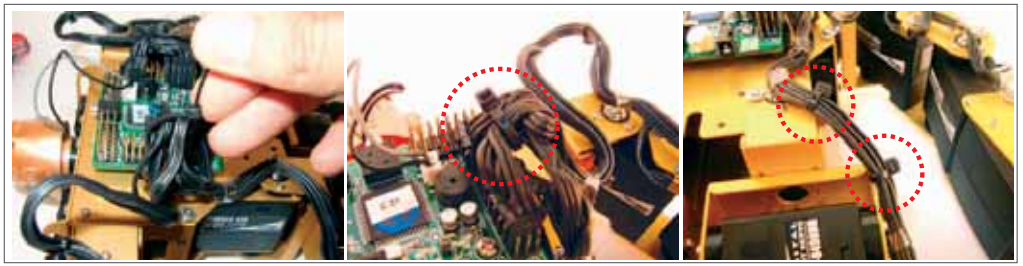
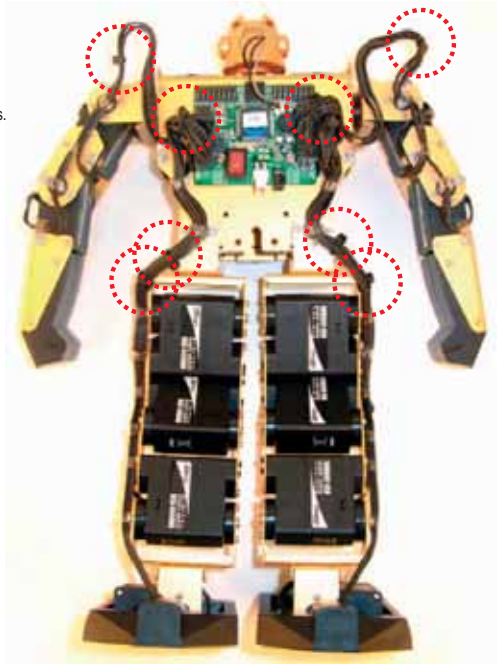
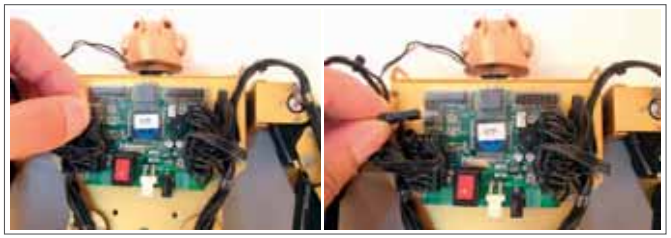


Imagen de los cables unidos por presillas.



(3) Montaje de la tapa de PINes y el LED.
Desconecte el servo izquierdo superior del controlador y coloque la tapa transparente sobre el controlador. Vuelva a conectar el servo.



Enchufe el conector del LED al controlador como se aprecia en la foto.

11. Montaje final del hardware.

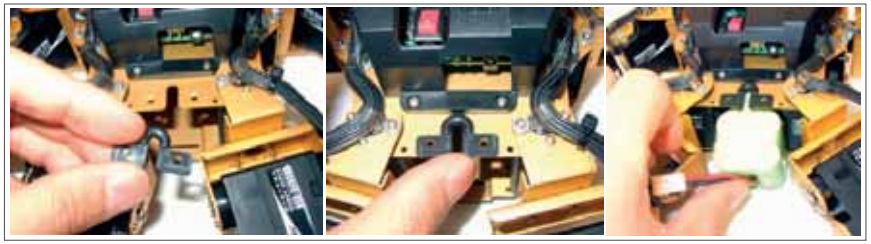
Ponga la tapa trasera del cuerpo sobre el controlador MR- C3024. La tapa trasera encaja con la delantera.
Estando seguro de no aprisionar ningún cable, ponga dos tornillos PH/T 2x26mm en la parte superior de la tapa y otros dos PH/M 2.6x4mm por la parte inferior.

Montaje de la tapa trasera finalizado.

(2) Instalación de las baterías

Prepare las baterías de 6V 1,000mAh Ni-MH, dos tornillos de 3X4mm, la parte inferior del cuerpo HR1B-0008, y el protector del cable de las baterías.

Primero, instale el protector del cable de las baterías en el cuerpo como aparece en la foto.
Coloque la batería en la parte inferior del cuerpo del Robot. Fíjese en la orientación del pack.



Coloque la tapa de las baterías y fíjelas con dos tornillos 3x4 rosca chapa.

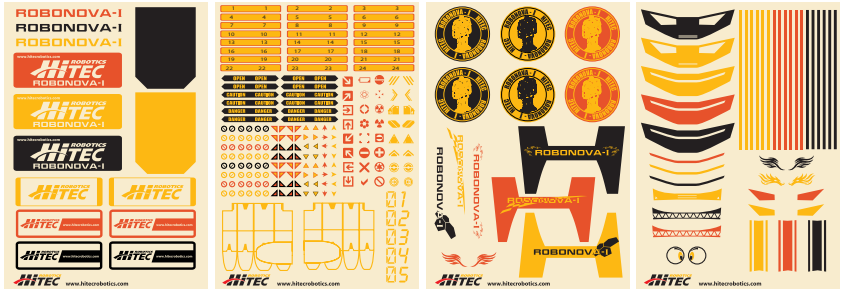
Conecte la batería al controlador.

Para cargar la batería, conecte el cargador al controlador y enchufe el cargador a una toma de corriente.

El cargador indica mediante un LED rojo que está cargando. Cuando termina el LED se enciende en verde.

(3) Terminando

Cuando esté terminado, el RoboNova-1 debe parecerse al de la fotografía.



El kit Robonova-1 kit incluye 4 juegos diferentes de pegatinas para personalizar su robot.

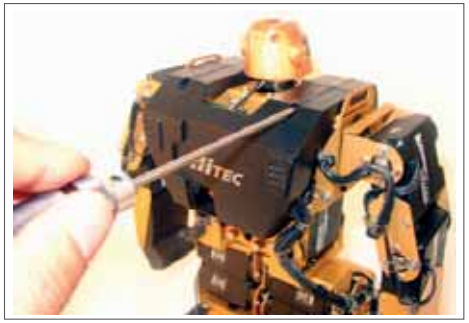
También se incluyen pegatinas para numerar los cables.

12. Instalando el Remocon y el sensor IR.

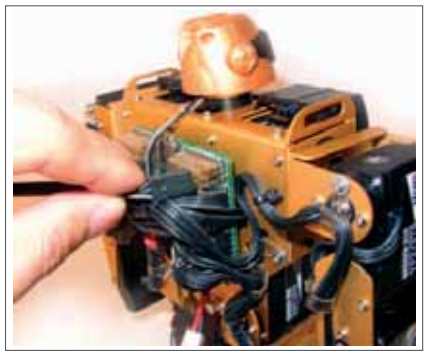
- No todos los kits incluyen el Remocon y el sensor IR.

El ROBONOVA-I puede ser controlado con los exclusivos Remocon y el sensor IR. Para instalar el sistema Remocon, el sensor IR y algo de cinta adhesiva de doble cara serán necesarios

Quite la tapa trasera del cuerpo del RN-I.

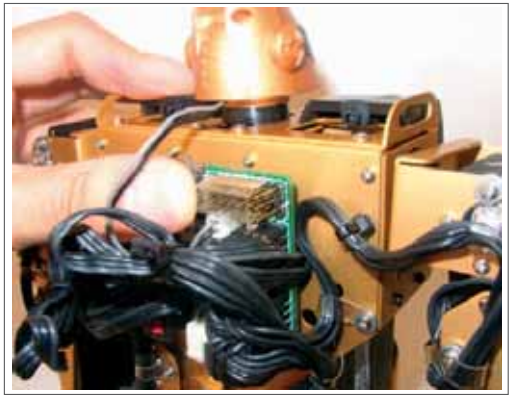


Conecte el conector del sensor IR en el puerto "AD7" ubicado en la parte superior derecha del controlador MR-C3024.

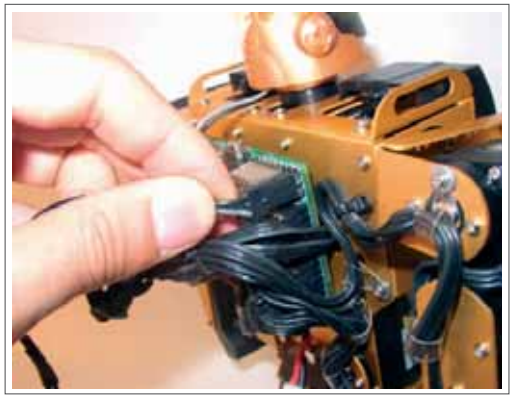


Antes de conectar el conector, quite la tapa transparente del controlador MR-C3024 y recorte el hueco necesario.

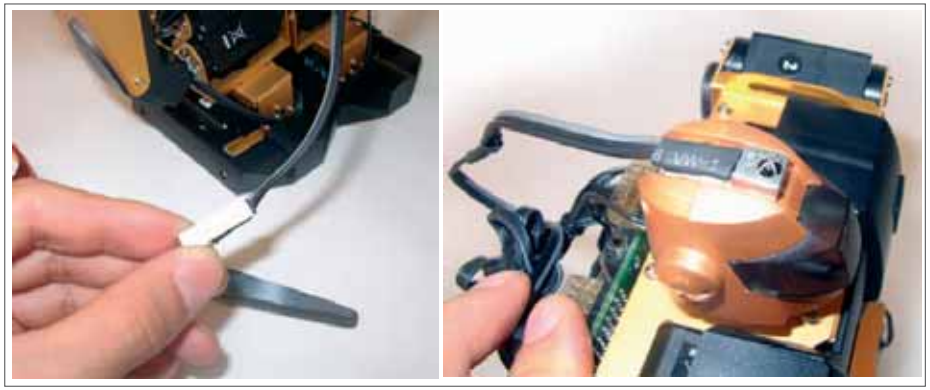
Vuelva a colocar la tapa transparente.



Conecte el sensor IR al MR-C3024. El cable gris del conector debe quedar hacia abajo.



Ponga un trozo de cinta adhesiva de doble cara 15X8mm en la parte trasera del sensor IR. Instale el sensor IR en la parte superior del RN-I.



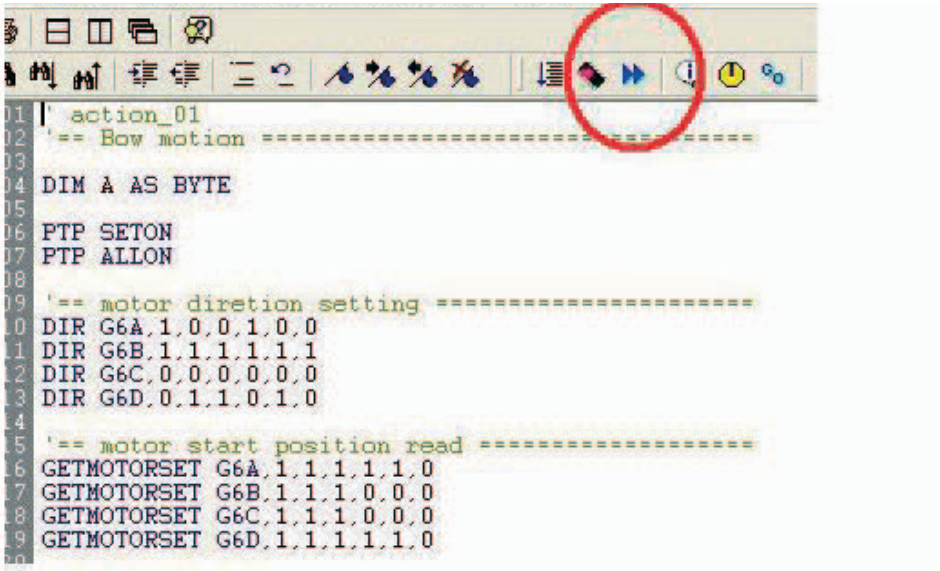
Vuelva a colocar la tapa trasera del cuerpo del RN-I.

El siguiente paso es la configuración de la programación del Remocon. Abra roboBASIC, y cree un nuevo programa, o cargue el fichero action_auto.bas de la carpeta template del CD.
En el programa de ejemplo, busque 'A = REMOCON(O) y cámbielo por A = REM0C0N(1)

```
MAIN1:
A = REMOCON(1)
A = A - ID
ON A GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K16,K17,K18,K19,K20
GOTO MAIN
'-----------------------------------------------
```

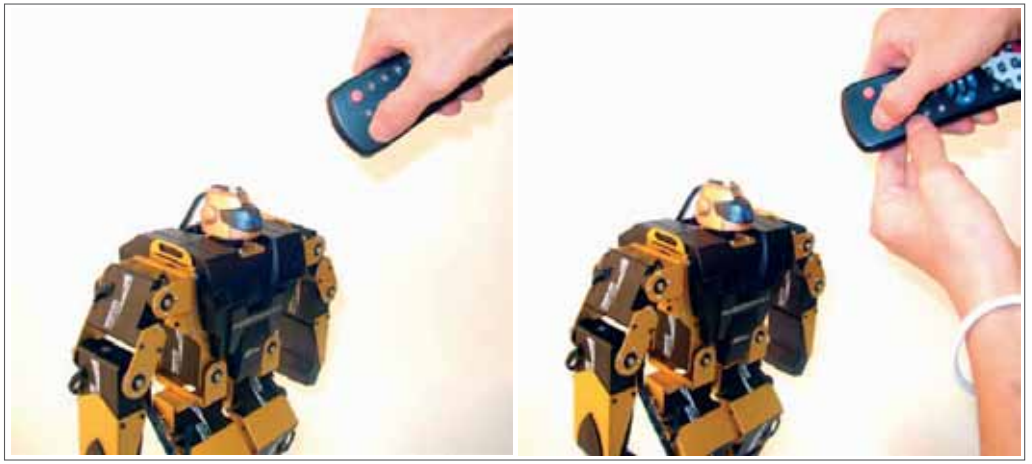
Conecte el ROBONOVA-I al PC con el cable para puerto serie.

Pulse sobre el botón "Run All" (Incluye la ejecución) para cargarlo en el controlador.



```
' action_01
'== Bow motion ================================

DIM A AS BYTE

PTP SETON
PTP ALLON

'== motor diretion setting ====================
DIR G6A,1,0,0,1,0,0
DIR G6B,1,1,1,1,1,1
DIR G6C,0,0,0,0,0,0
DIR G6D,0,1,1,0,1,0

'== motor start position read ================
GETMOTORSET G6A,1,1,1,1,1,0
GETMOTORSET G6B,1,1,1,0,0,0
GETMOTORSET G6C,1,1,1,0,0,0
GETMOTORSET G6D,1,1,1,1,1,0
```

El Remocon se puede identificar con un ID entre 1 y 4. Esto permite el control de hasta 4 robots ROBONOVA-I por remocons diferentes, de manera simultanea y sin interferencias.
Encienda el ROBONOVA-I y con el Remocon apuntando al sensor IR, pulse la tecla "P1" situada en la parte superior izquierda del Remocon.
Después, pulse el número de ID (uno de los botones "1~4"). Deje pulsado los dos botones durante 2 segundos.
Cuando cambie las pilas, el ID puede perderse ID.



IV. Preparativos para manejar el ROBONOVA-I

* RoboBasic debe estar instalado antes de manejar el ROBONOVA-I
1 Vea "(2) Instalación de RoboBasic' en el apartado "1. Instalación y ejecución de RoboBasic".

1. Configuración de movimientos básicos y aplicaciones.
(1) Configuración inicial de RoboBasic para el ROBONOVA-I
Es muy importante configurar el software RoboBasic para que funcione perfectamente.
1) Configuración de puertos para el controlador y el PC. Elección de controlador. ' Vea "(3) Ejecución de RoboBasic" en el apartado "1. Instalación y ejecución de RoboBasic". El controlador se configurará automáticamente para el ROBONOVA-I.
2) Explicación de los ajustes de los neutros en Robobasic. ' Vea "(5) Configuración de neutros en RoboBasic" en el apartado "1. Instalación y ejecución de RoboBasic".

(2) Lista de los comandos que aparecen en el programa plantilla.
1) Comando "goto AUTO" (Manual de comandos de RoboBasic, p44 en el CD)

goto AUTO

Salta al programa plantilla.

Sintaxis

- Comando : goto AUTO

Explicación del comando

Comando que lanza el programa plantilla.

Ejemplo

Goto AUTO 'Salta al programa plantilla.

Nota: Este comando solo es para el controlador MR-C3024 y el Robot Robonova-I.

```
GOTO AUTO
FILL 255,10000


DIM RR AS BYTE
DIM A AS BYTE
DIM A16 AS BYTE
DIM A26 AS BYTE

CONST ID = 0      ' 1:0, 2:32, 3:64, 4:96

'** Action command check (50 - 82)
IF RR > 50 AND RR < 83 THEN GOTO action_proc

RR = 0

PTP SETON
PTP ALLON

'** motor direction setting ********************
DIR G6A,1,0,0,1,0,0
DIR G6B,1,1,1,1,1,1
DIR G6C,0,0,0,0,0,0
DIR G6D,0,1,1,0,1,0
```

(El comando "Goto AUTO" aparece al principio del programa plantilla).
Abra el fichero "Overall Template Program,bas" en la carpeta Template del CD de RoboBasic
-> Consulte 1, programas de plantilla en la sección VI, Aplicaciones.

Haga click sobre "RUN ALL" (ejecución integrada) para cargar el fichero en el controlador.

Al finalizar la carga, el altavoz emitirá un sonido y el ROBONOVA se pondrá de pié.
- Si el ROBONOVA no se queda bien firme, tendrá que ajustar los puntos neutros de los servos.
' Consulte (5) Ajustes de los puntos neutros en RoboBasic en la sección V. Como usar el software con el ROBONOVA-I
- Si hubiese muchas diferencias entre la posición de firmes del ROBONOVA y la fotografía inferior, compruebe que los servos, los horns y las pletinas están bien montadas.
- Después de comprobar el montaje, encienda el interruptor. Si la posición es la misma que la de las fotos (Consulte (5) Ajustes de los puntos neutros en RoboBasic), podrá realizar las 29 acciones básicas con el programa plantilla. (Consulte la tabla REMOCON para la asignación de teclas.)
Compruebe la configuración del software de RoboRemocon o del IR REMOCON (Consulte (4) Uso de ROBOREMOCON con RoboBasic en el apartado 1. Instalación de RoboBasic).

2) Tabla para la configuración de remocom y acciones.
- Los movimientos configurados en el MR-C3024 se pueden manejar con el programa RoboRemocon o con el IR REMOCON (no disponibles en algunas versiones del kit).
(Consulte el apartado "11. Instalación del IR REMOCON" para la instalación y configuración del IR REMOCON) (Consulte el apartado "3. Instalación y configuración de RoboRemocon" acerca del uso del software RoboRemocon software)

Tabla de asignación de teclas de REMOCON

ACTION	key	Motion	Variable	Code
0	power	ON : motor on->Basic Position OFF: Sitting Position-> motor off	A16	16
1	1	Bow -> Basic Position		1
2	2	Raise arms -> Basic Position		2
3	3	Sit -> Basic Position		3
4	4	Sit -> Raise arms -> Basic Position		4
5	5	Raise a leg -> Basic Position		5
6	6	Spread the legs -> Extend arms -> Right-left tilt -> Basic Position		6
7	7	Flap arms like a bird		7
8	8	Kick		8
9	9	Handstand		9
10	0	Walk fast		10
11	*	Left turnabout		22
12	#	Right turnabout		24
13	▲	Forward		11
14	◀	Left move		14
15	■	Sit(->)Stand up	A26	26
16	▶	Right move		13
17	▼	Reverse		12
18	△	Front tumbling		21
19	◁	Left cartwheel		28
20	□	Front attack		29
21	▷	Right cartwheel		30
22	▽	Rear tumbling		31
23	A	Left attack		15
24	B	Right attack		20
25	C	Left front jab		17
26	D	Right front jab		27
	E,F,G	27,28,29 Spare 18,32,23		18

V. Manual de software del ROBONOVA-I
1. Instalación y manejo de roboBASIC v2.5

Marca registrada
Windows es una marca registrada de Microsoft Corporation

Nota

Este manual es para la instalación y manejo de roboBASIC. Esta versión es diferente de las anteriores, e incluyendo el manual, se pueden hacer cambios sin previo aviso para mejorar el producto.
RoboBasic es un software bajo licencia, lo que hace imposible su reproducción, publicación, transmisión y/o distribución tanto del manual como del software sin autorización.
HITEC Robotics http://www.hitecrobotics.com

(1) Acerca de roboBASIC
RoboBASIC está basado en el lenguaje de programación BASIC y está diseñado específicamente para el control de los controladores de la serie MR-C para gestión de Robots.
RoboBASIC es un lenguaje educativo que mejora el lenguaje de programación BASIC para permitir el control de Robots.
RoboBASIC es compatible con MS Windows 98, ME, 2000 y XP

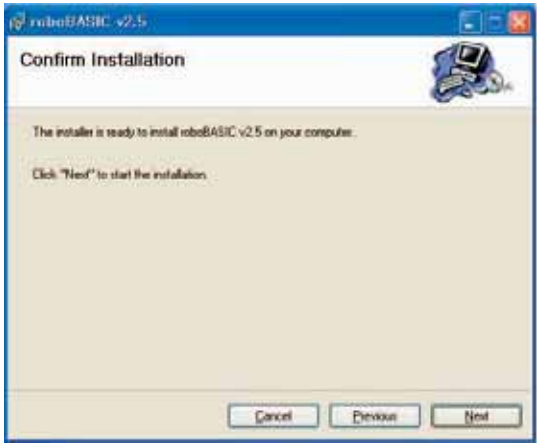
(2) Instalación de roboBASIC
- El software de RoboBASIC puede instalarse desde el diskette o el CD incluidos con el robot RN-1, o descargado desde la página Web de HITEC Robotics (http://www.hitecrobotics.com)

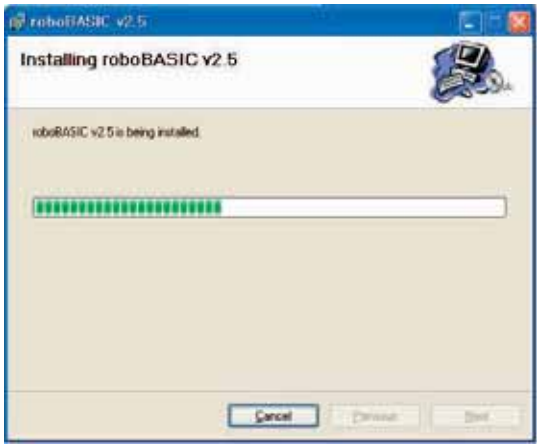
'Al instalar ROBOBASIC, ROBOSCRIPT y R0B0REMOCON se instalan automáticamente.

roboBASIC v2.5
Welcome to the roboBASIC v2.5 Setup Wizard
The installer will guide you through the steps required to install roboBASIC v2.5 on your computer.
Click "Next" to continue.
WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.
Cancel
Next

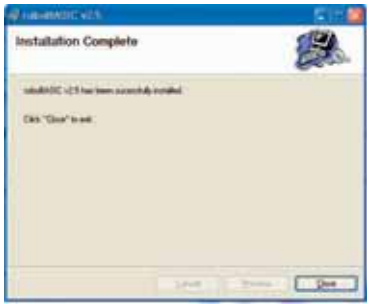
Pulse sobre "SETUP.EXE" para comenzar la instalación.

Seleccione la carpeta de instalación de ROBOBASIC





Comienza la instalación de ROBOBASIC

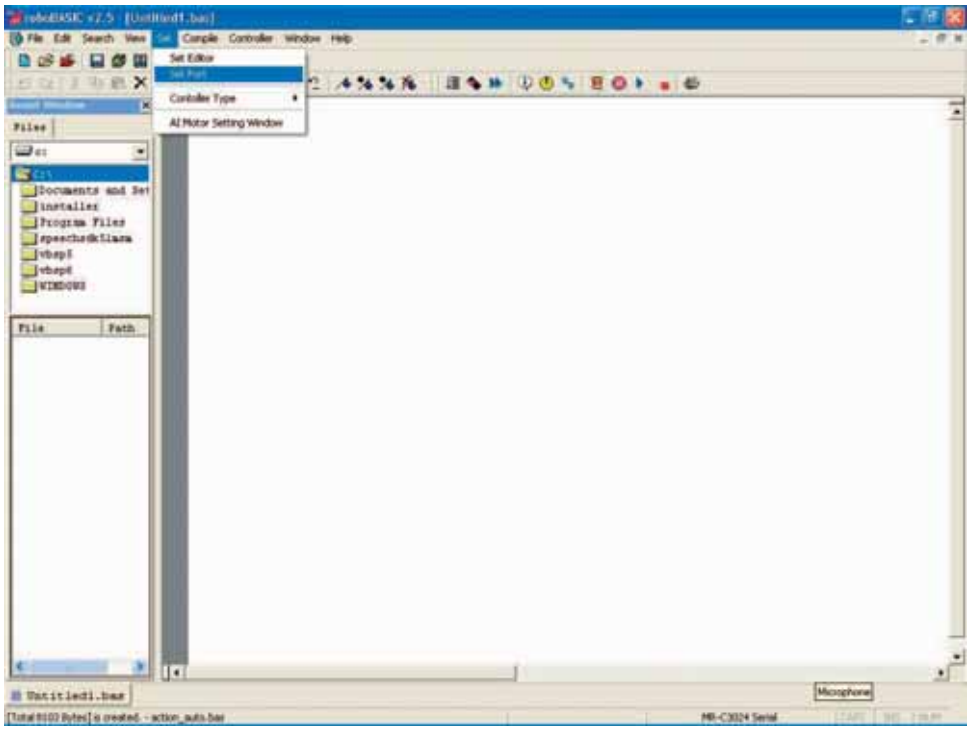


Instalación terminada

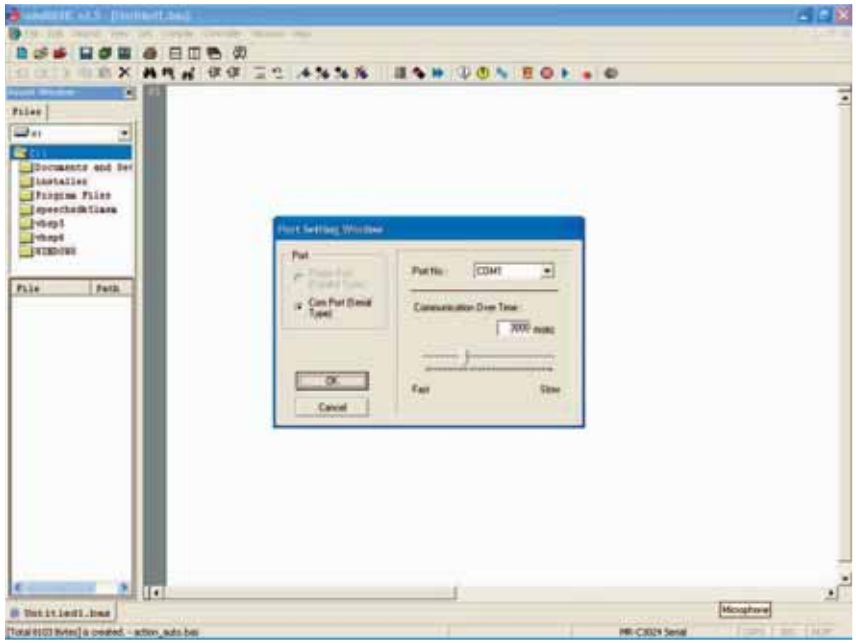
(3) Configuración inicial de roboBASIC

Conecte el cable de datos y la alimentación al controlador, y encienda la unidad. Abra roboBASIC.

1 Seleccione el tipo de controlador.
Seleccione el controlador MR-C3000. Si dispone de un MR-C3024, selecciónelo.
roboBASIC v2.5 - [Untitledi.bas]

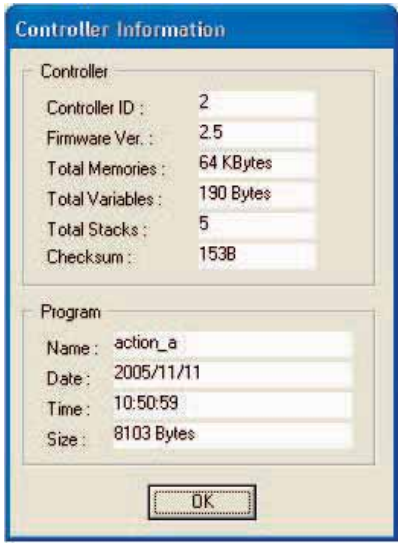


2 Elección del puerto serie. Seleccione el puerto serie que vaya a utilizar.
Para confirmar que puerto serie usará, abra el Administrador de Dispositivos de Windows.
[Panel de control -" Sistema -* Administrador de dispositivos -* puertos (COM y LPT)]



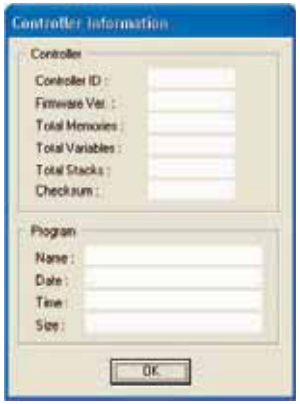
Su usa roboBASIC por primera vez, se deberá configurar el controlador y el sistema. Una vez terminada la configuración, no tendrá que repetirlas, a no ser que haga cambios en el sistema. Para comprobar que el controlador se comunica con el PC, abra la ventana de Información sobre el Controlador.

Pulse sobre [Controller(C) -o Controller Information o sobre el icono <ICON>.
Si se establece la comunicación, se mostrarán los datos del controlador en la ventana "controller information".



Si hubiese algún error en la comunicación, aparecerá un mensaje de error. Pulse el botón OK. La ventana "controller information" no contendrás ningún dato.

(Mensaje de error del controlador)

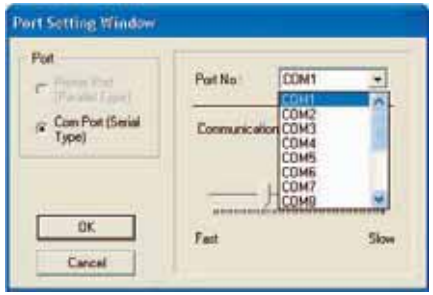


(La ventana no muestra ningún dato del controlador)

Vuelva a comprobar que el cable de datos está bien conectado y que el controlador recibe alimentación.
Repita el proceso de configuración.

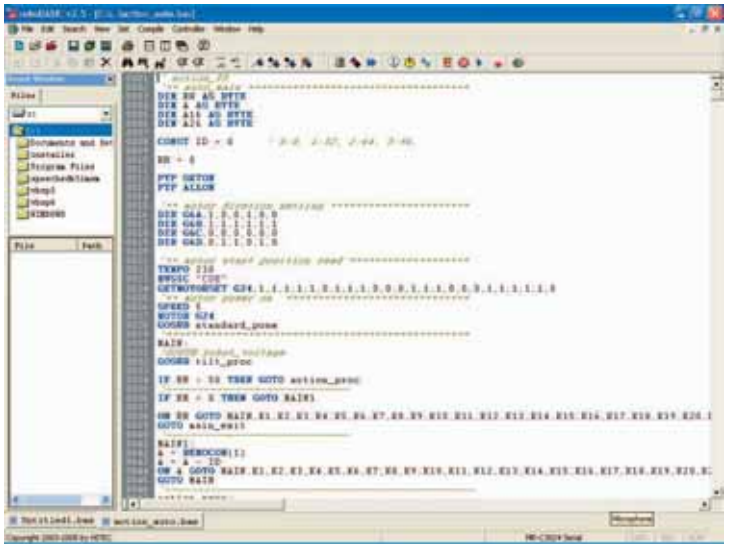
Haga click sobre [Set up (T) - Port set up

Puede cambiar el puerto si lo necesita.



(4) Programación en RoboBASIC

Cuando haya terminado la instalación y la configuración, podrá comenzar con la programación.
Cree un nuevo programa, o abra uno existente. Todos los ficheros se guardan con la extensión [*.bas].

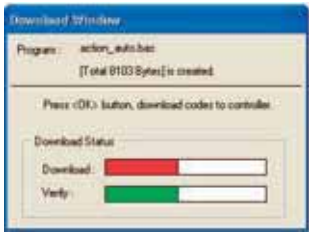


Para crear un programa, nuevo o cargado anteriormente, en el controlador, debe convertirlo en código objeto compilándolo. Sólo entonces podrá volcarlo en la ROM del controlador. La compilación y la transferencia se pueden hacer por separado, o en un solo paso.

Haga click sobre el icono "Make Object Code", (F2) o elija "Make Object Code" en el menú [Compile (0) - Make object code (F2)]

Haga click sobre el icono "Download" , (F6) o seleccione Download en el menú. [File (0) -* Download (F6)]

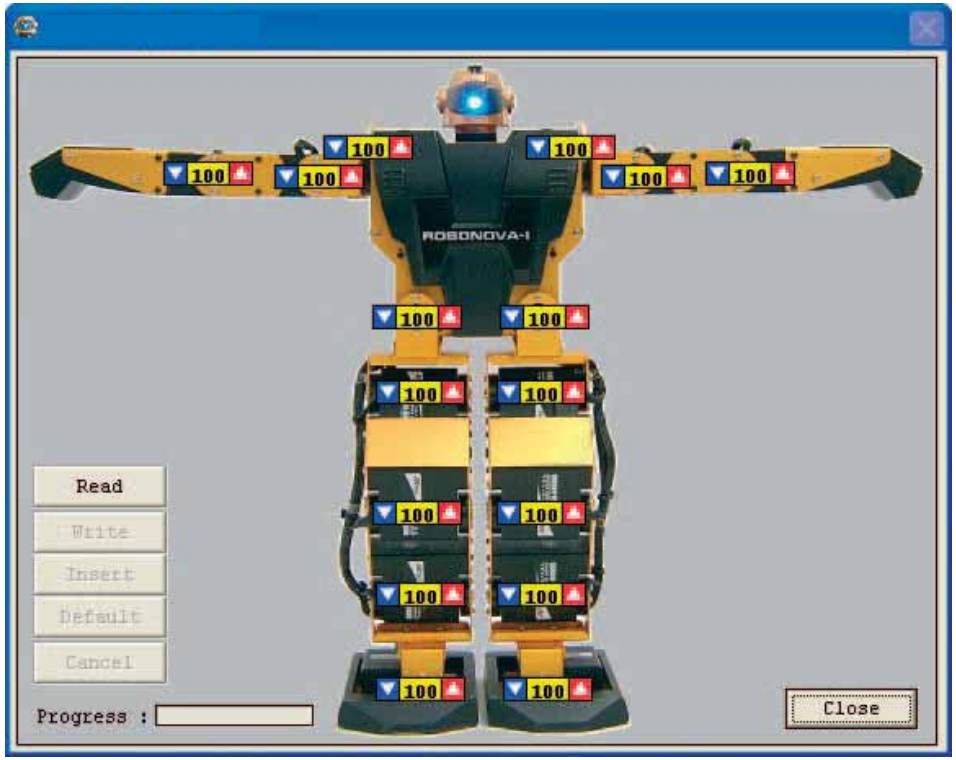
Haga click sobre el icono "Integrated Execution" , (F9) o seleccione Run All en el menú [File (0) - Run All (F9)]



(5) Configuración de los puntos neutros en RoboBasic

Incluso habiendo montado perfectamente el Robot, los servos pueden estar algo descentrados. Es necesario ajustar los puntos neutros de los servos en su posición correcta.
Los puntos neutros se controlan con el programa RoboBasic.

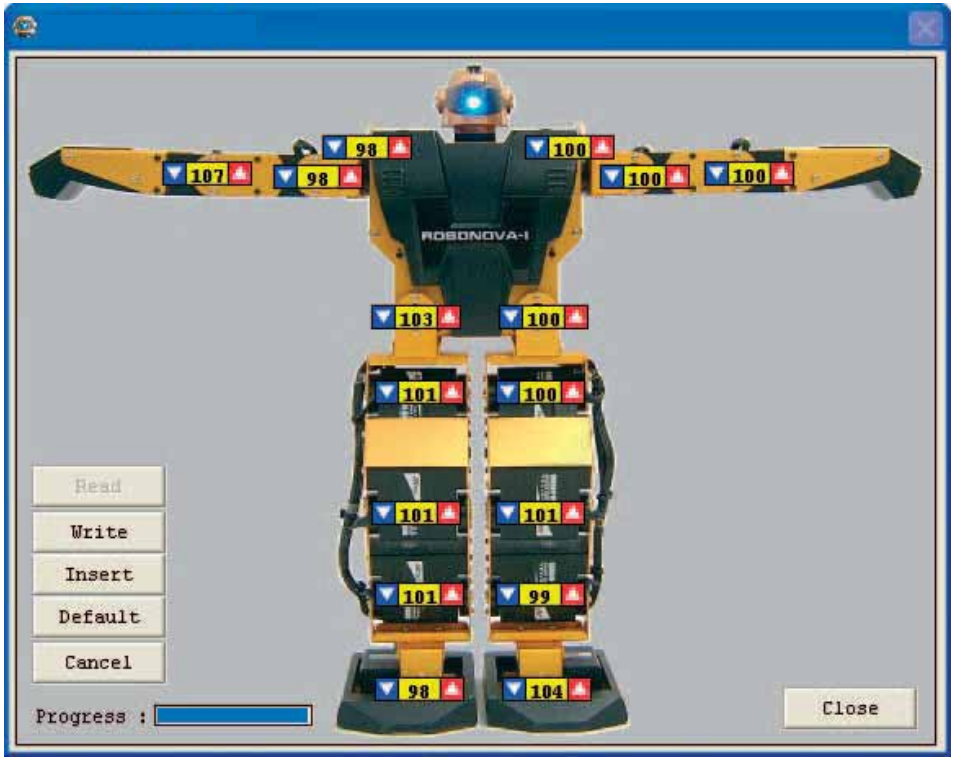
- Para acceder a la configuración de los puntos neutros en RoboBasic, vaya a ' Compile ' Set Zero Point.



- Se abrirá una pequeña ventana con la imagen del ROBONOVA. Al lado de cada servo, aparece el punto neutro. (Vea la foto del RN-I). Para que el Robot se sitúe en la misma posición que en la foto, pulse sobre el botón "Read".

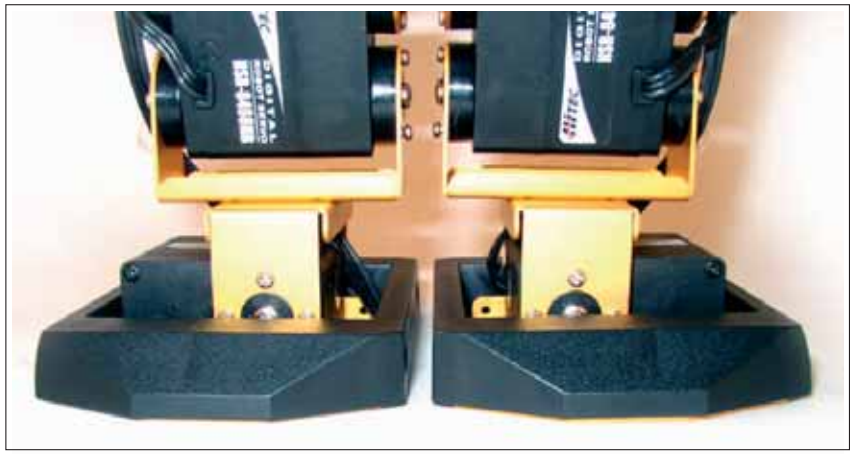
* (Aviso) Al pulsar el botón "READ", las piernas y brazos del robot se moverán para adoptar la posición de la imagen. Si sostiene el robot en sus manos, podría hacerse daño. Le sugerimos que antes de pulsar sobre el botón "READ" deposite el robot sobre una superficie plana.

ROBONOVA-I

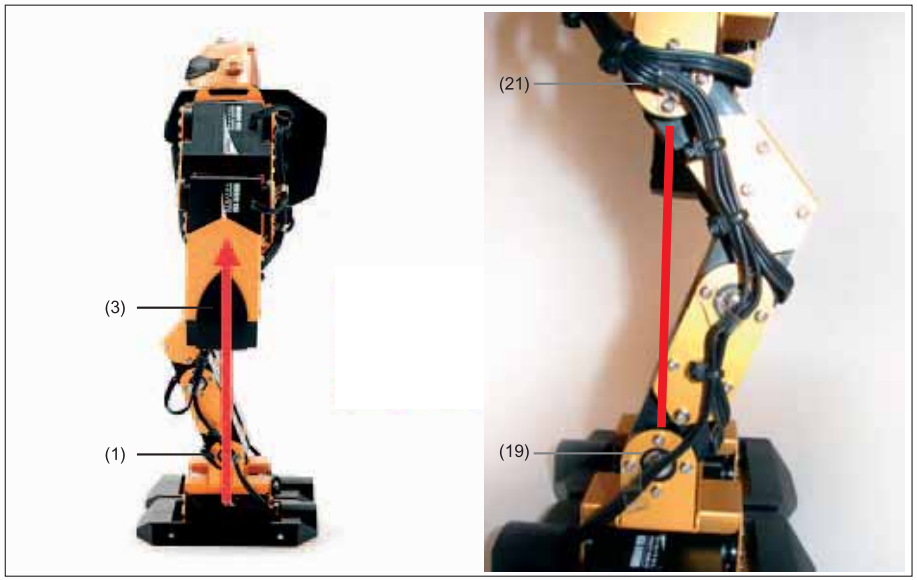


(Puede ajustar los valores de los puntos neutros pulsando sobre las flechas)

Puntos neutros adecuados.



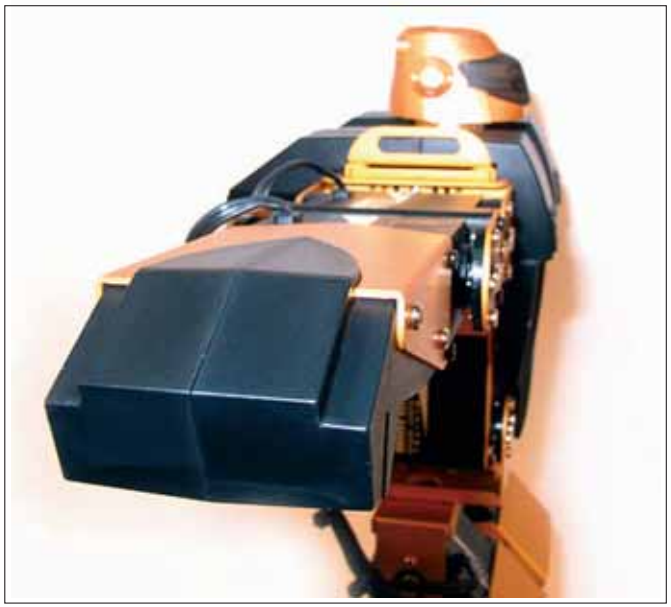
Los dos pies deben estar apoyados completamente sobre la superficie.



Los servos No.1 y No.3 de la pierna izquierda (No.19 y No.21 de la derecha) deben estar en línea recta.

ROBONOVA-I

La separación entre ambas piernas debe ser de unos 8mm.



Los brazos deben estar horizontales como en la foto superior.

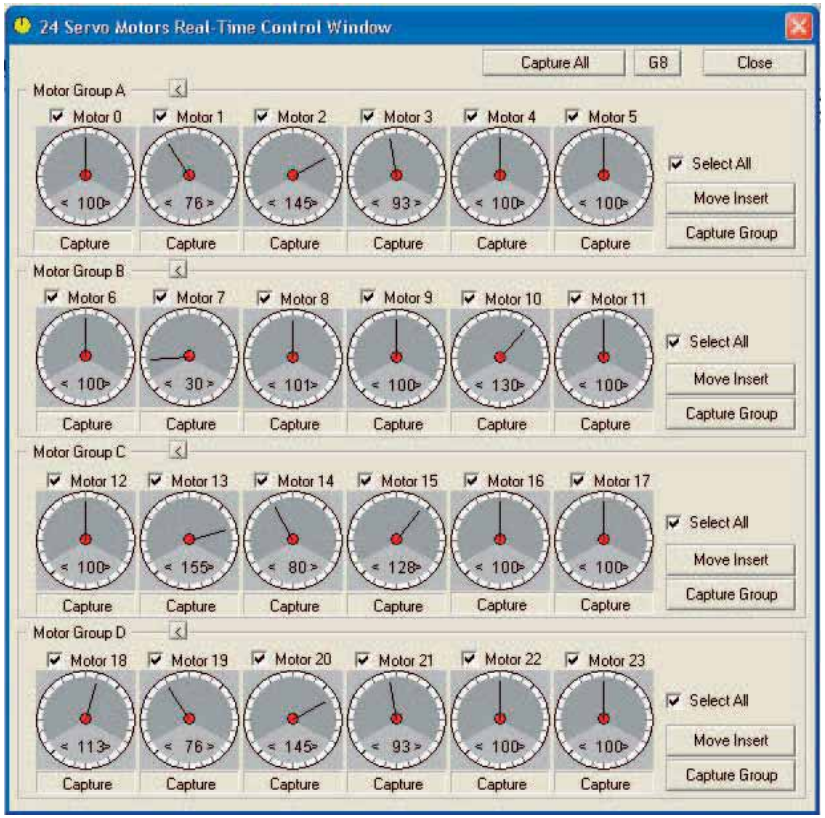
- Puede ajustar los puntos neutros usando las flechas arriba y abajo.
Para guardar la configuración en RoboBasic, haga click sobre el botón "Write". Esto insertará los nuevos valores en el programa. Para enviar los nuevos valores al robot de manera inmediata, haga click en el botón "Insert".
Esto enviará los nuevos valores al controlador MR-C3024 sin tener que usar la función normal para cargar programas de RoboBasic. Para comprobar los nuevos valores,

(6) Control de servos en tiempo real

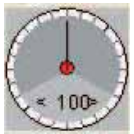
Con el control de servos en tiempo real, podrá controlar los movimientos del robot de manera fácil y rápida.

Conecte el controlador MR-C3024 con el PC y arranque el programa ROBOBASIC.

haga click sobre el icono control de servos en tiempo real o pulse sobre [Controller(C) ' Servo motor realtime control (F7)]


24 Servo Motors Real-Time Control Window
Capture All
G8
Close
Motor Group A
Motor 0 Motor 1 Motor 2 Motor 3 Motor 4 Motor 5
100 76 145 93 100 100
Capture
Select All
Move Insert
Capture Group
Motor Group B
Motor 6 Motor 7 Motor 8 Motor 9 Motor 10 Motor 11
100 30 101 100 130 100
Motor Group C
Motor 12 Motor 13 Motor 14 Motor 15 Motor 16 Motor 17
100 155 80 128 100 100
Motor Group D
Motor 18 Motor 19 Motor 20 Motor 21 Motor 22 Motor 23
113 76 145 93 100 100

Servo a controlar

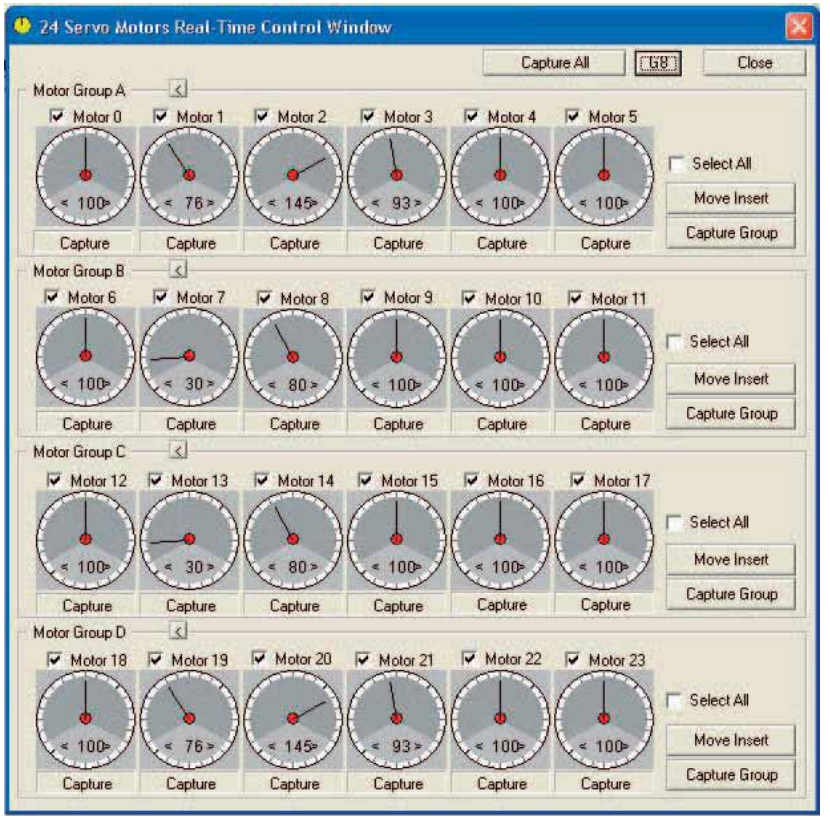


La posición del servo oscila entre 10~190

Despliega o cierra el grupo de servos.

ROBONOVA-I

Agrupa los 24 servos en grupos de seis.

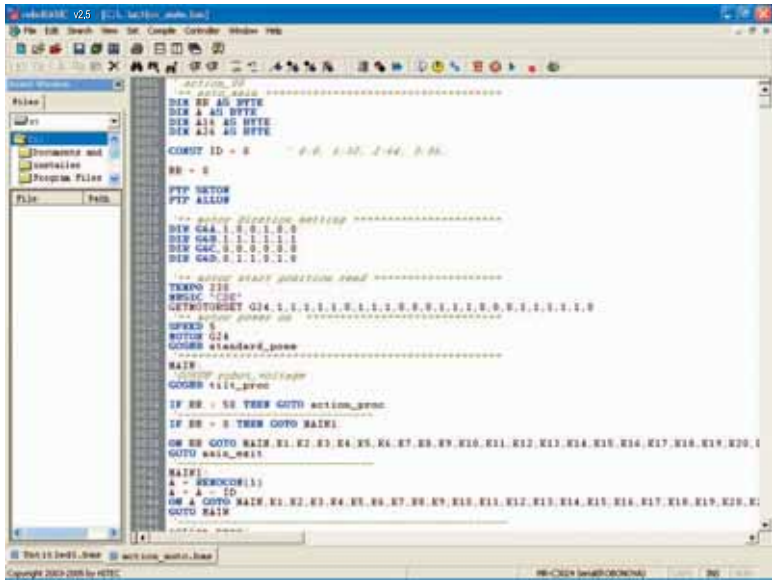

24 Servo Motors Real-Time Control Window
Capture All
G8
Close
Motor Group A
Motor 0 Motor 1 Motor 2 Motor 3 Motor 4 Motor 5
100 76 145 93 100 100
Capture
Select All
Move Insert
Capture Group
Motor Group B
Motor 6 Motor 7 Motor 8 Motor 9 Motor 10 Motor 11
100 30 80 100 100 100
Motor Group C
Motor 12 Motor 13 Motor 14 Motor 15 Motor 16 Motor 17
100 30 80 100 100 100
Motor Group D
Motor 18 Motor 19 Motor 20 Motor 21 Motor 22 Motor 23
100 76 145 93 100 100

Inserta el comando "MOVE" en la ventana de programación.
ex) MOVE G8A, 100, 100, 100, 100, 100, 100, 100, 100

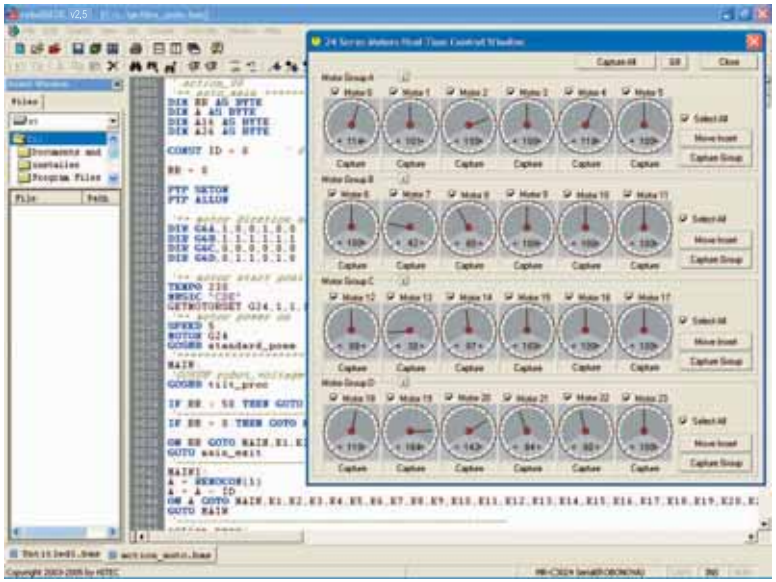
Cierra el control de servos en tiempo real.

(7) RoboBasic ROBONOVA-I. Control de servos

1) Control de servos en tiempo real.



(Conecte el ROBONOVA al PC usando el cable de datos y accione el interruptor. Situé el cursor en la línea del programa donde quiera insertar los comandos "MOVE"



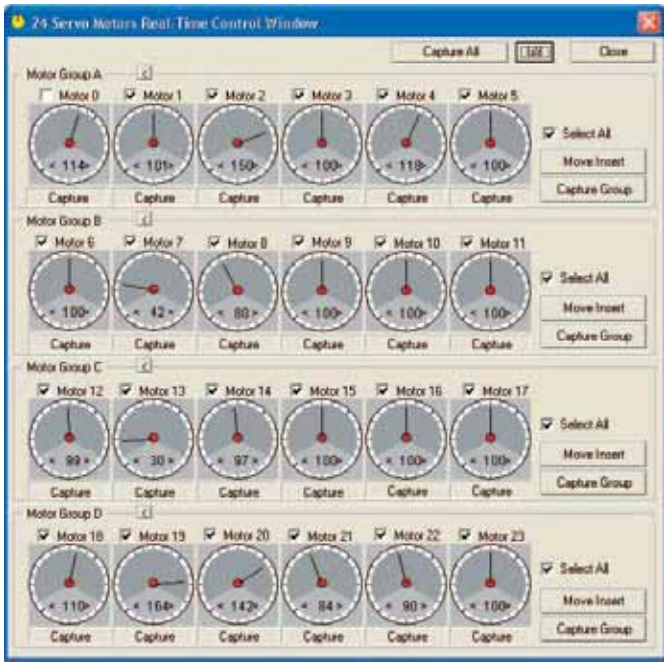
(Haga click sobre el menú ' controller ' Control de servos en tiempo real. Se abrirá la venta de control de servos.)

- Hay dos maneras de usar el control de servos en tiempo real. Uno es mover a mano los servos. El otro, es usar las flechas situadas junto a los dígitos, para aumentar o disminuir los valores.

1 Método de ajuste manual

- Quite la marca del servo #0 del grupo A. Esto hará que pueda mover libremente el servo.
- Lleve el servo a la posición deseada. Para guardar la posición, deje el servo en ese lugar y marque la casilla.
- Para insertar la posición en el código de RoboBasic, haga click sobre el botón "MOVE INSERT". Si cualquier casilla de verificación está en blanco, esa posición no se insertará en el código, solo se insertará un espacio.

2 Ajuste con el ratón

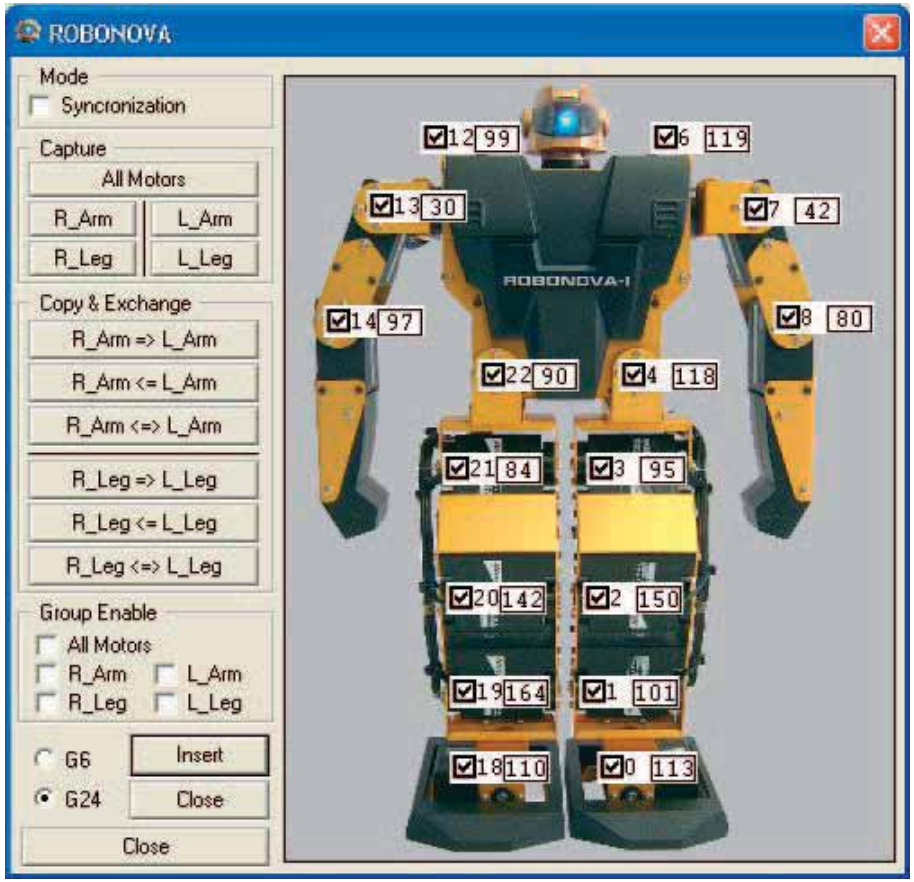


- Compruebe que la casilla esté marcada, y después mueva la aguja o las flechas izquierda/derecha del servo deseado . A medida que va cambiando, el servo se moverá. Una vez en la posición deseada, haga click sobre el botón "MOVE INSERT" para insertar los valores en RoboBasic.

Si quiere marcar todas las casillas de un grupo, haga click en el botón "Select All". Para quitar las marcas de todas las casillas de un grupo, click sobre el botón "Select All"

2) ROBONOVA-1 Control de servos.

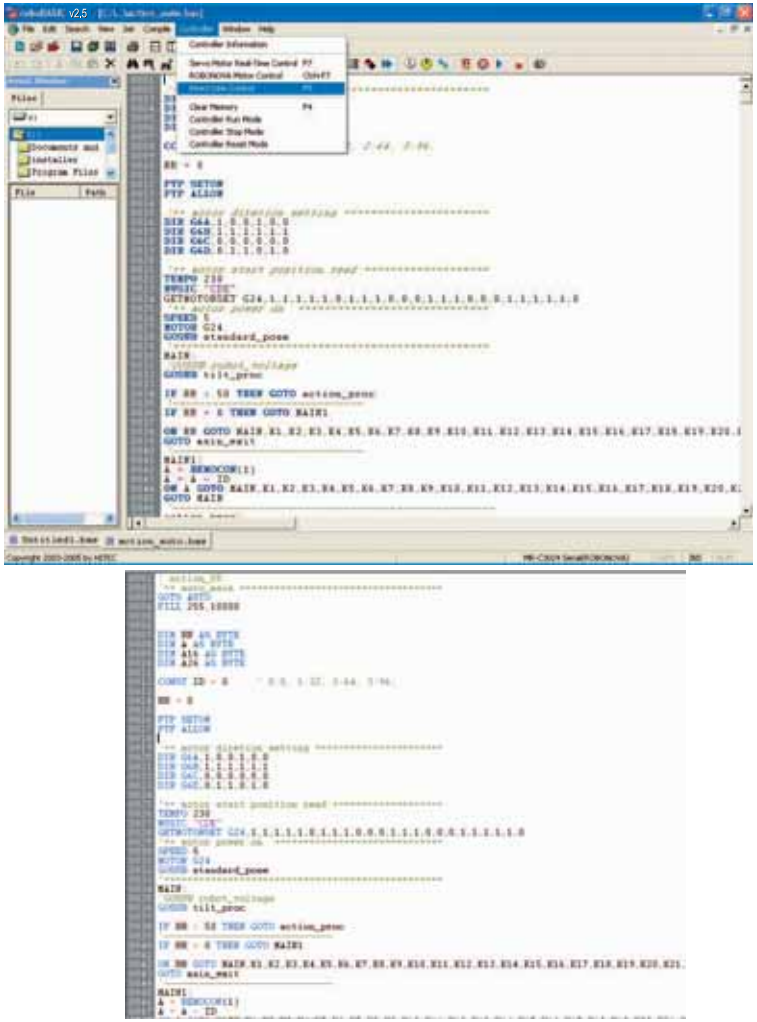
- También puede manejar los servos usando el control de servos del Robonova.
Menú' controller '



- Ventana de control de servos: Marque la casilla del número de servo/valor (ángulo del servo). Cada servo puede ser controlado pulsando sobre su casilla.

Control directo

- Se usa con una sola línea de código.



Menú ' Controller' Línea de control directo

Nota: Este comando se usa junto a los comandos OUT , MOVE, SERVO, MOVE, POSE, MOVEpose.

- Seleccione la línea a ejecutar y haga click sobre "Direct line control".

2.Uso de roboScript v2.S

Marca registrada
Windows es una marca registrada de Microsoft Corporation

Nota

Este manual es para la instalación y manejo de RoboScript. Esta versión es diferente de las anteriores, e incluyendo el manual, se pueden hacer cambios sin previo aviso para mejorar el producto.
RoboScript es un software bajo licencia, lo que hace imposible su reproducción, publicación, transmisión y/o distribución tanto del manual como del software sin autorización.

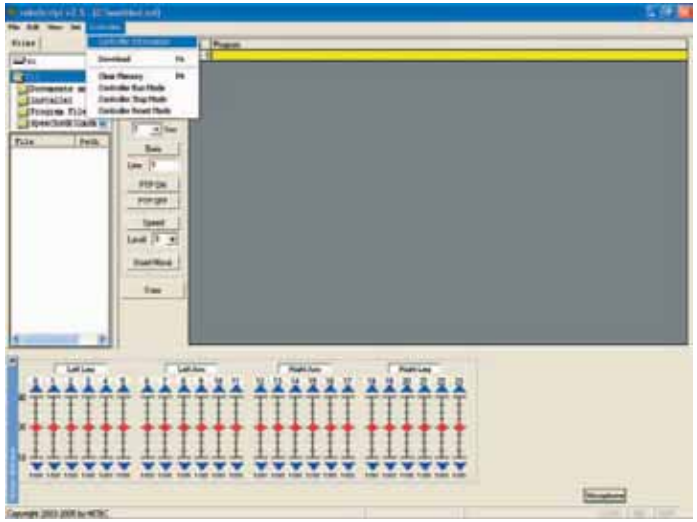
HITEC Robotics http://www.hitecrobotics.com

(1) Acerca de RoboScript

RoboScript es un entorno de scripts desarrollado para simplificar la modificación y utilización de RoboBASIC. RoboScript es compatible con MS Windows 98, ME, 2000 y XP

(2) Configuración inicial de roboScript

Conecte el cable de datos al controlador, y encienda la unidad. Abra RoboScript. Aparecerá la ventana de configuración inicial. Para comprobar que RoboScript se haya instalado correctamente, compruebe la información del controlador. Click sobre [Controller(C) -" Controller Information] o el icono



Si se establece la comunicación, se mostrará la información sobre el controlador en la ventana "controller information".

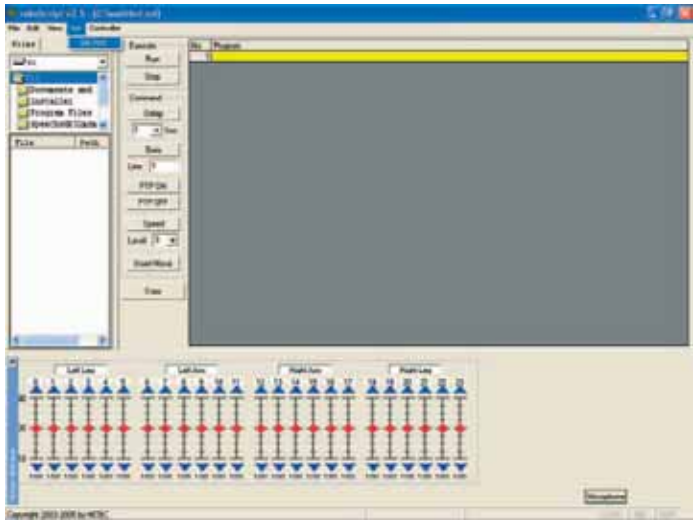
(Se muestra la información sobre el controlador)

Si hubiese cualquier fallo en la comunicación, se mostrará un mensaje de error. Pulse sobre el botón OK. La ventana "controller information" no mostrará ninguna información en este caso.

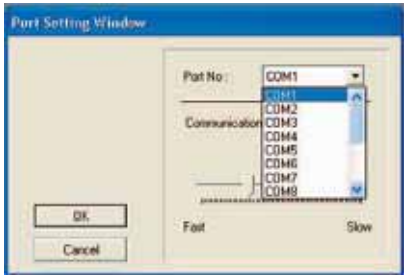
<Controller error message>

(la ventana no contiene información sobre el controlador)

Vuelva a comprobar que el cable de datos está bien conectado y que el controlador recibe alimentación. Repita el proceso de configuración. Click [Setup (T) - Port setup]

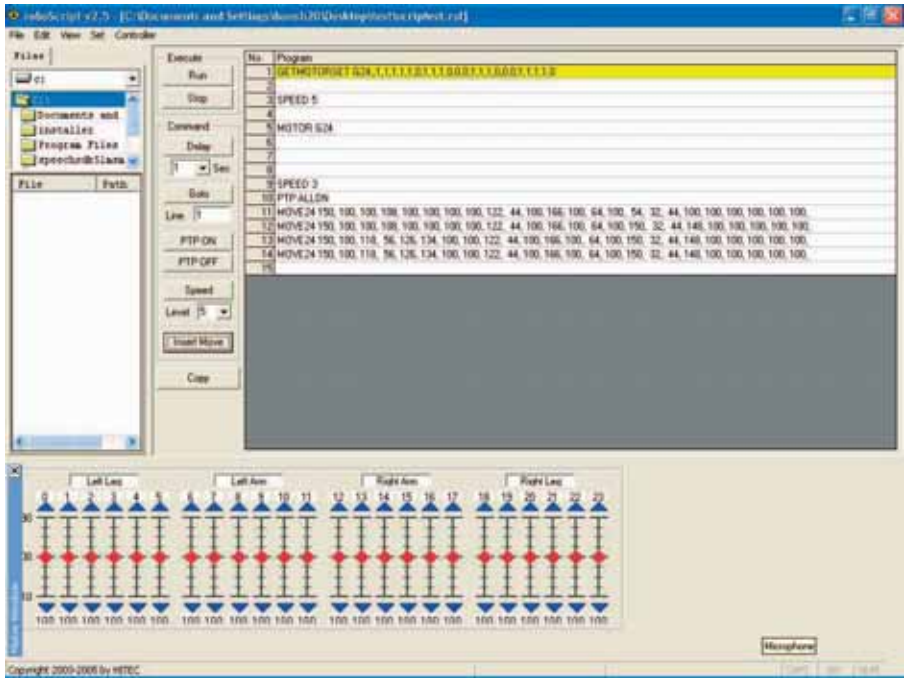


Puede cambiar el puerto si fuese necesario.

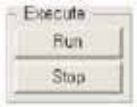


(3) Programación en RoboScript

Cuando haya terminado la instalación y la configuración, podrá comenzar con la programación.
Cree un nuevo programa, o abra un fichero existente. Todos los ficheros se guardan con la extensión [*.rsf].



En RoboScript, se puede comprobar el movimiento del robot sin tener que enviar un programa al controlador. Si hace falta, se puede modificar y volver a "copiar" a Robobasic.



Ejecuta/Detiene el programa creado/modificado.

Delay - Hace una pausa de 0.5seg a 30seg.

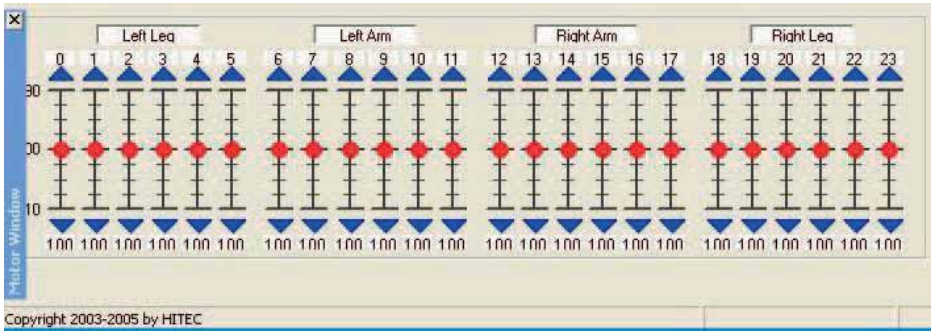
Goto - Inserta un comando goto en el código

PTP - Activa/desactiva la función PTP.

Speed - Establece la velocidad del servo entre 1 y 15.

Insert Move - Inserta un comando "Move" en el código.

Copy - Copia el código al portapapeles para que pueda pegarlo en Robobasic con [Ctrl+v] o [Pegar]


Left Leg
Left Arm
Right Arm
Right Leg
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Motor Window
Copyright 2003-2005 by HITEC

Ventana de control de servos - Cada servo puede ser controlado en tiempo real arrastrando el indicador de las barras deslizantes. Al terminar de posicionar el servo, lo guarda en el código con la calve "Insert Move".

(3) Uso de RoboRemocon V2.5

Marca registrada
Windows es una marca registrada de Microsoft Corporation

Nota

Roborecom es un software bajo licencia, lo que hace imposible su reproducción, publicación, transmisión y/o distribución tanto del manual como del software sin autorización.

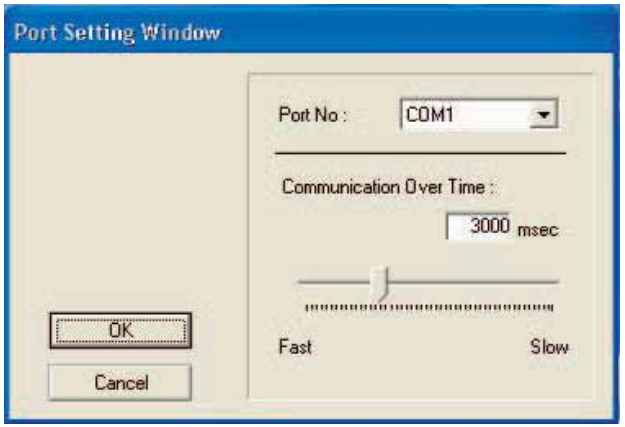
HITEC Robotics http://www.hitecrobotics.com

(1) Acerca de RoboRemocon

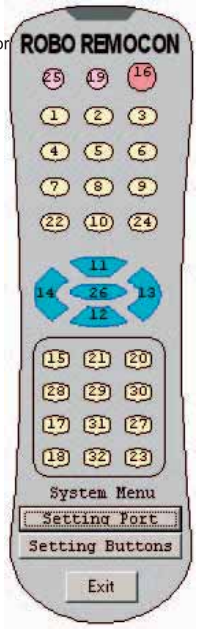
RoboRemocon permite el control sin cables del robot. Para que pueda manejar RoboRemocon, necesitará un programa RoboScript creado previamente. El control remoto no es necesario para comprender el funcionamiento del programa RoboRemocon.
RoboRemocon es compatible con MS Windows 98, ME, 2000 y XP

(2) Programación y manejo de RoboRemocon

Para comprobar si RoboRemocon está bien configurado, compruebe la información del controlador

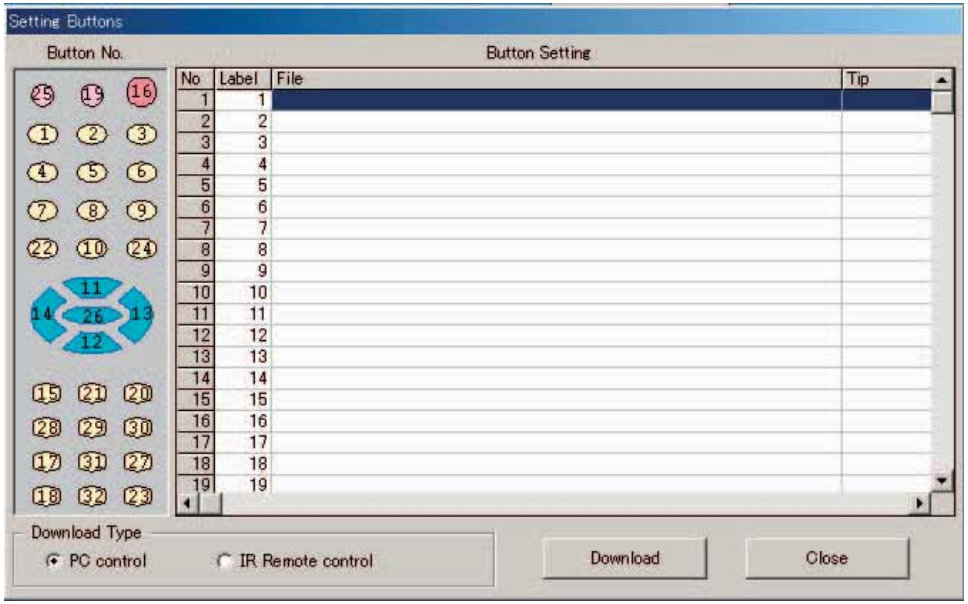


Configuración del puerto de comunicaciones

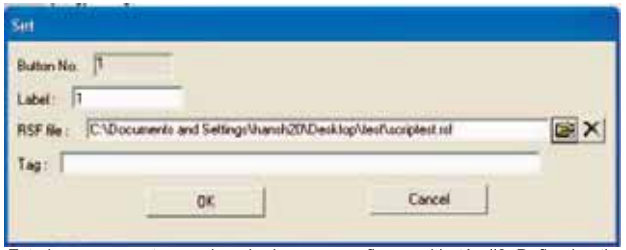


Pulse sobre el botón

Pulse sobre el botón



Pulse dos veces sobre la tecla del remocon y establezca la función.



Esta imagen muestra un ejemplo de como configurar el botón #0. Define la etiqueta, el fichero RoboScript y el tag del botón #0

En el menú Download Type, haga click sobre "PC Control" si quiere usar el ROBOREMOCON del PC.

Haga click sobre "IR Remote control" si quiere usar el control remoto IR Remocon.

Carga un programa en el controlador.

Sale del programa

(3) Como usar RoboRemocon en RoboBasic

- Abra el fichero "Overall Template Program.bas".

```
' action_00
'== auto_main ====================================
DIM RR AS BYTE
DIM A AS BYTE
DIM A16 AS BYTE
DIM A26 AS BYTE

CONST ID = 0      ' 0:0, 1:32, 2:64, 3:96,

RR = 0

PTP SETON
PTP ALLON

'== motor direction setting ======================
DIR G6A,1,0,0,1,0,0
DIR G6B,1,1,1,1,1,1
DIR G6C,0,0,0,0,0,0
DIR G6D,0,1,1,0,1,0

'== motor start position read ====================
TEMPO 230
MUSIC "CDE"
GETMOTORSET G24,1,1,1,1,1,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,1,1,0
'== motor power on  ==============================
SPEED 5
MOTOR G24
GOSUB standard_pose
'==================================================
MAIN:
'GOSUB robot_voltage
GOSUB tilt_proc

IF RR > 50 THEN GOTO action_proc
'-------------------------------
IF RR = 0 THEN GOTO MAIN1

ON RR GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15
GOTO main_exit
'-------------------------------
MAIN1:
A = REMOCON(1)
A = A - ID
ON A GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,
```



```
DIM RR AS BYTE                  // Se declara RR como variable del ROBOREMOCON.
RR =0                           // Inicializa RR a 0

MAIN:

                                //Salta a MAIN1 puesto que RR = 0
IF RR> 50 THEN GOTO action_proc
IF RR=O THEN GOTO MAIN1

MAIN1:
A = REMOCON(1)
A = A - ID
ON A GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8-
GOTO MAIN
```

Puesto que la variable RR se inicializa a 0, se usará el HITEC REMOCON.
Cuando se recibe una señal desde el RoboRemocon, el valor de RR cambiará haciendo que el programa salte a la sub-rutina MAIN.

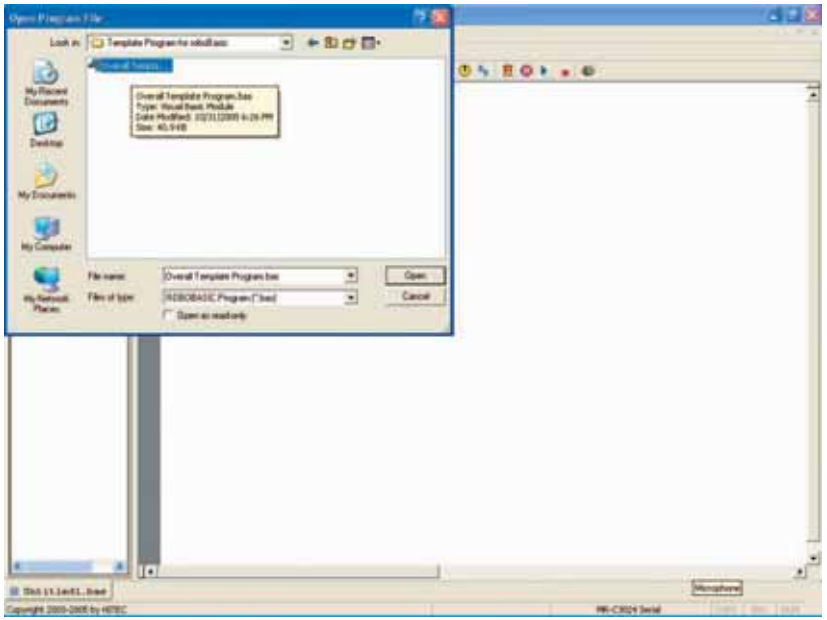
```
MAIN:
'GOSUB robot_voltage
GOSUB tilt_proc

IF RR > 50 THEN GOTO action_proc
'-------------------------------
IF RR = 0 THEN GOTO MAIN1

ON RR GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12
GOTO main_exit
'-------------------------------
```

Procesa cada valor (K1.K2---.) con ON RR GOTO.
(Vea la tabla de funcionamiento del IR REMOCON)

VI. Aplicaciones del programa

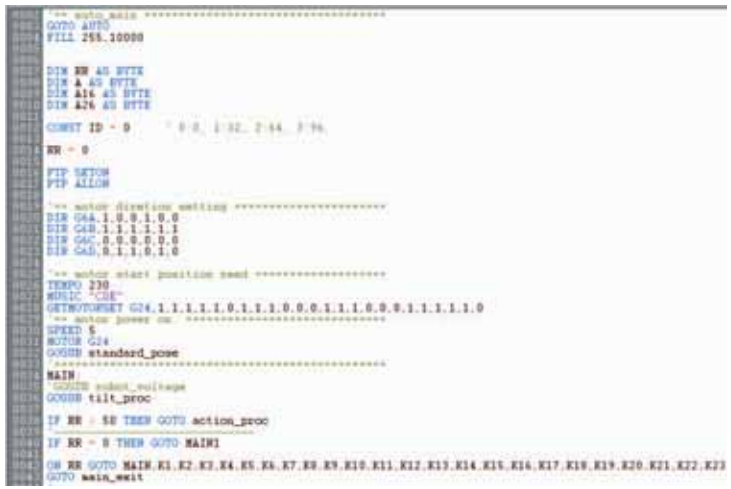
1. Instalación y uso del programa plantilla

(1) Instalación del programa plantilla
Abra el fichero "Overall Template Program.bas" de la carpeta "Template Program for roboBASIC" incluida en el CD.



(2) Formas de modificar el programa plantilla.
El programa se divide en 3 partes:

1) Prográma básico




```
GOTO AUTO

FILL 255,10000          ' Guarda el programa plantilla en la dirección 10000 de la memoria del controlador MR-C3024

Ex)
GETMOTORSET G24.1,1,1,1,1,1,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,1,1,0
SPEED 5
MOTOR G24

ACTION 1
DELAY 1000
ACTION 2

DIM RR AS BYTE          'La primera variable declarada en el programa se asigna al RoboRemocon.

CONST ID =0             'El HITEC IR REMOCON (con función ID) puede manejar 4 remocons y 4 ROBONOVAs
                        simultáneamente y sin interferencias.
                        1 EX: CONST ID =1 (ID No.1)

IF RR > 50 AND RR < 83 THEN GOTO action_proc
                        'Si el valor de entrada está entre "50 y 83" al estar controlador por RoboRemocon, salta a
                         "action_proc"
Para otros comandos usados en el ejemplo, consulte la lista de comandos.
```

2) MAIN"

```
'-----------------------------------------------------
MAIN:
'GOSUB robot_voltage
GOSUB tilt_proc

IF RR > 50 THEN GOTO action_proc
'-----------------------------------------------------
IF RR = 0 THEN GOTO MAIN1

ON RR GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K16,K17,K18,K19,K20,K21,
GOTO main_exit
'-----------------------------------------------------
MAIN1:
A = REMOCON(1)
A = A - ID
ON A GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K16,K17,K18,K19,K20,K21,K
GOTO MAIN
'-----------------------------------------------------
action_proc:
A = RR - 50
ON A GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K16,K17,K18,K19,K20,K21,K
RETURN
'-----------------------------------------------------
```

MAIN es una función muy importante en un programa ROBONOVA-I.

El comando 'GOSUB robot_voltage" se evalúa en la primera línea de la sub-rutina MAIN.

Este comando es para los avisos por voltaje bajo.
(Vea el apartado VI. Programación de aplicaciones' 2. ROBOBASIC programming application '(7) Configuración de los avisos por bajo voltaje)

Para usar el comando, borre el carácter (') y se activará la función de aviso por bajo voltaje (6.25v)

```
GOSUB robot_tilt            'Si el robot cae, este comando hará que se levante (necesita el sensor de caídas).

IF RR = 0 THEN GOTO MAIN1   'Si RR no recibe ninguna señal, salta a MAIN1.

MAIN1:
A = REMOCON(1)
A = A - ID
ON A GOTO MAIN,
K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K16,K17,K18,K19,K20,K21,K22,K23,K24,K25,K26,K27,K28,K29,K30,K31
,K32
GOTO MAIN
                            'MAIN1: es la rutina que permite al HITEC REMOCON controlar el robot.
                            'Los valores que llegan desde el remocon se almacenan en la variable A. El valor de ID es
                            distinto. La sentencia, A=A-ID permite los movimientos de cada tecla del remocon.
```

3) Funciones de movimientos y teclas de remocon

```
0064 '================================================
0065 k1:
0066     GOSUB bow_pose
0067     GOSUB standard_pose
0068     GOTO main_exit
0069 k2:
0070     GOSUB hans_up
0071     DELAY 500
0072     GOSUB standard_pose
0073     GOTO main_exit
0074 k3:
0075     GOSUB sit_down_pose
0076     DELAY 1000
0077     GOSUB standard_pose
0078     GOTO main_exit
0079 k4:
0080     GOSUB sit_hans_up
0081     DELAY 1000
0082     GOSUB standard_pose
0083     GOTO main_exit
0084 k5:
0085     GOSUB foot_up
0086     GOSUB standard_pose
0087     GOTO main_exit
0088 k6:
0089     GOSUB body_move
0090     GOSUB standard_pose
0091     GOTO main_exit
0092 k7:
0093     GOSUB wing_move
0094     GOSUB standard_pose
0095     GOTO main_exit
0096 k8:
0097     GOSUB right_shoot
0098     GOSUB standard_pose
0099     DELAY 500
0100     GOSUB left_shoot
0101     GOSUB standard_pose
```


4) Movimiento individual

```
0234     RETURN
0235 '================================================
0236 sit_down_pose16:
0237     IF A16 = 0 THEN GOTO standard_pose16
0238     A16 = 0
0239     SPEED 10
0240     MOVE G6A, 100, 151,  23, 140, 101, 100
0241     MOVE G6D, 100, 151,  23, 140, 101, 100
0242     MOVE G6B, 100,  30,  80, 100, 100, 100
0243     MOVE G6C, 100,  30,  80, 100, 100, 100
0244     WAIT
0245 '== motor power off  ============================
0246     MOTOROFF G24
0247     TEMPO 230
0248     MUSIC "FEDC"
0249     RETURN
0250 '================================================
0251 standard_pose16:
0252     TEMPO 230
0253     MUSIC "CDE"
0254     GETMOTORSET G24,1,1,1,1,1,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,1,1,0
0255 '== motor power on  =============================
0256     MOTOR G24
0257     A16 = 1
0258 '================================================
0259     SPEED 10
0260     GOSUB standard_pose
0261     RETURN
0262 '================================================
0263 '================================================
0264 bow_pose:
0265     MOVE G6A, 100,  58, 135, 160, 100, 100
0266     MOVE G6D, 100,  58, 135, 160, 100, 100
0267     MOVE G6B, 100,  30,  80,   ,   ,
0268     MOVE G6C, 100,  30,  80,   ,   ,
0269     WAIT
0270     DELAY 1000
0271     RETURN
```

Para asignar un movimiento individual a una tecla del remocon, haga la operación e insértela en el código de la rutina de la tecla a asignar.

Ex: Crear una acción individual y asignarla a una tecla. La siguiente rutina hace que el LED parpadee.

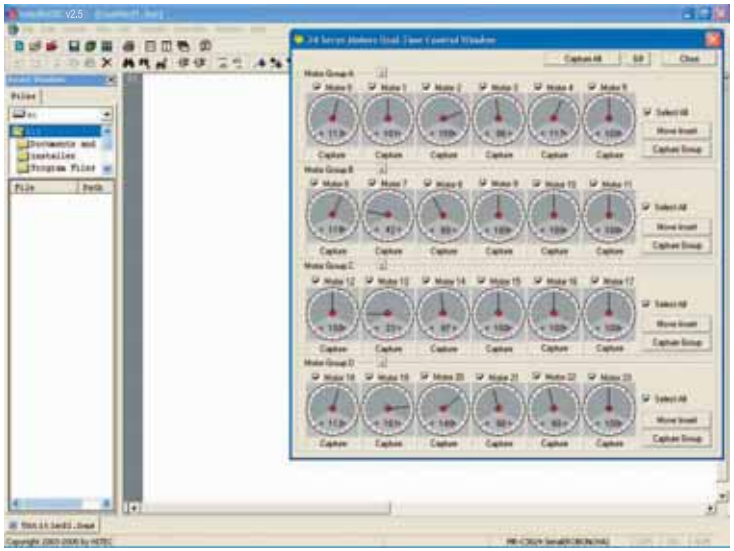
```
LED_TOGGLE:
        OUT 52,1
        DELAY 1000
        OUT 52,0
        DELAY 1000
```

Una vez creada la rutina, se asigna a una tecla del remocon.
Borre el contenido de la sub-rutina k18 del fichero "action_auto.bas". Inserte los siguientes comandos.

```
K18:
GOSUB LED_TOGGLE
GOTO main_exit
```

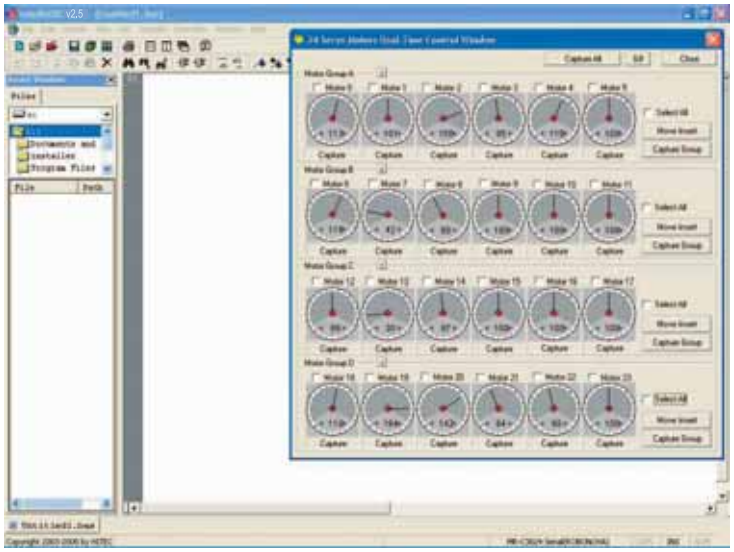
Haga click sobre el botón "Run All" (conlleva ejecución) para cargarlo en el controlador. Pulse la tecla 'E' del remocon y la luz comenzará a parpadear.

2. Programación de aplicaciones



(Esta imagen es un ejemplo del control de los ángulos para cada grupo de motores)

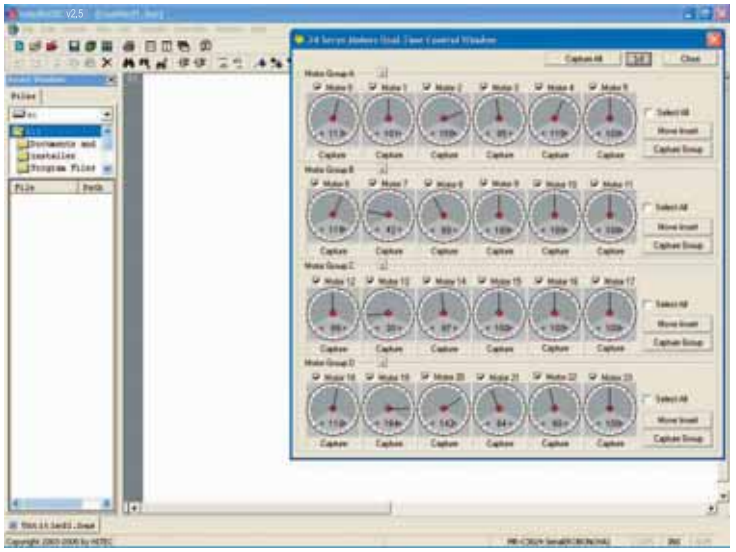
- Encienda el robot después de conectar el controlador al PC.
- Menú ' Compile ' Control de servos en tiempo real
- Se mostrarán los valores actuales de los ángulos de los servos.

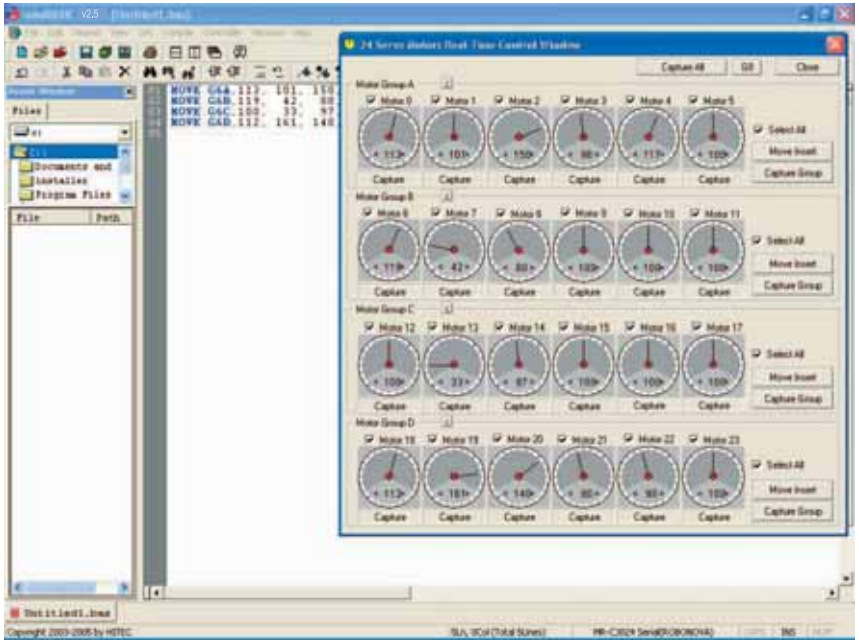


(Para crear una acción, quite las marcas del servo, o servos, utilizados en la acción. Para deseleccionar todos los servos de un grupo, haga click en el botón "Select All".)

ROBONOVA-I

(Ahora, los servos están desbloqueados. Mueva los servos a la posición deseada y vuelva a marcar las casillas. Se guardará la posición, y los servos volverán a estar bloqueados.)

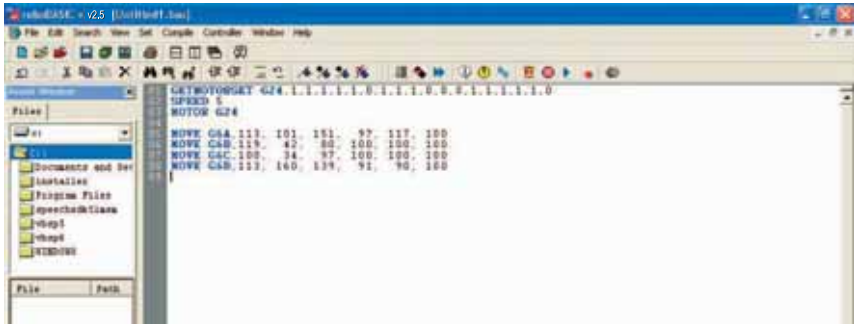




(Click sobre "motion insert" para insertar el movimiento en RoboBasic.)

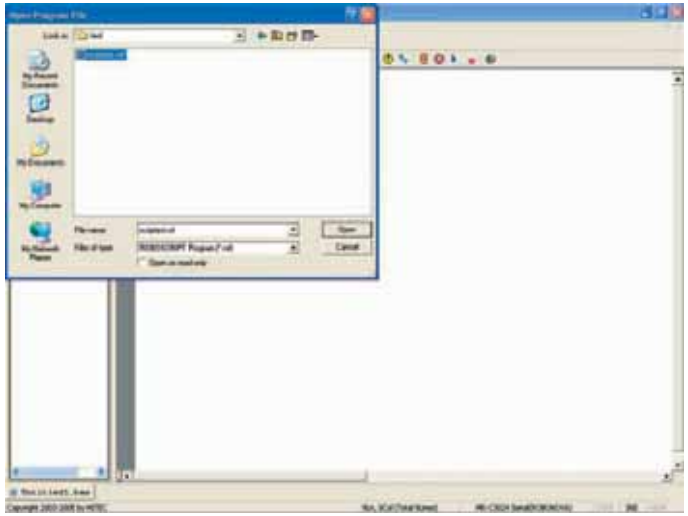
Para reproducir la acción definida, debe insertar los comandos siguientes justo antes. Los puede copiar desde el programa plantilla o teclearlos directamente.

G24.1,1,1,1,1,1,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,1,1,1,0 SPEED 5 MOTOR G24



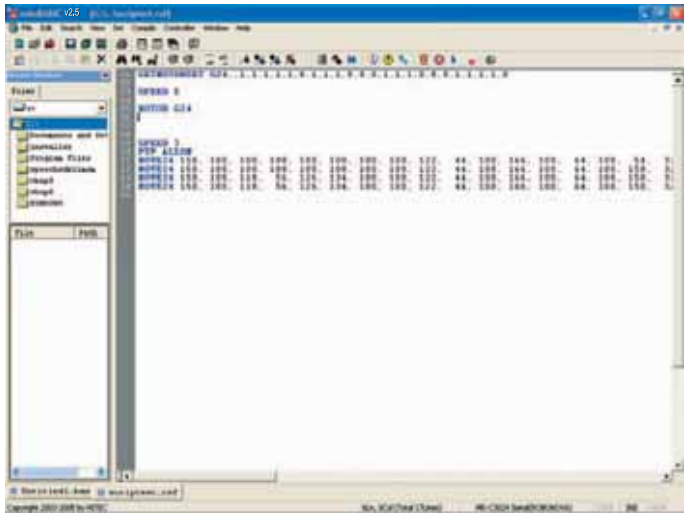
Cuando termine,haga click en el botón "Run ALL" (conlleva ejecución) y se enviará la acción al controlador. El robot reproducirá la acción definida.

(2) Como importar un fichero ROBOSCRIPT GETMOTORSET



(Abra un fichero *.rsf)

- Abra un fichero RoboScript (*.rsf) en RoboBasic.



(Comandos insertados en el programa RoboBasic)

Añada las siguientes instrucciones antes de los comandos RoboScript importados.

GETMOTORSET G24,1,1,1,1,1,1,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,1,1,1,0
SPEED 5
MOTOR G24

Haga click sobre "Run ALL" (con lleva ejecución) para enviarlo al controlador.

(3) AD Conversion

El comando AD se explica de nuevo en el "Apartado 11, explicación de comandos RoboBasic para el proceso de señales analógicas" del manual incluido en el CD

AD()

Convierte una señal analógica recibida por el puerto AD en una señal digital.

Sintaxis

AD ([Puerto AD])

Explicación del comando

Hay ocho puertos AD, numerados del 0 al 7, en los controladores de la serie MR-C3000
(Puertos digitales E/S 32 a 39), que convierte una señal analógica de los sensores externos, o dispositivos, en señales digitales.

Ejemplo del comando

En el siguiente ejemplo, se envía una valor al módulo LCD después de recibir una señal analógica desde el puerto AD No. 1.

```
DIM a AS BYTE           Declara la variable "a" de tipo byte.
LCDINIT                 Inicializa el uso del módulo LCD.
CLS                     Se borran todos los datos en el visor LCD.
CSOFF                   Se oculta el cursor.
MAIN:                   Se declara una etiqueta llamada MAIN.
a = AD (1)              Se guarda en la variable "a" el valor recibido por el puerto AD #1.
LOCATE 5,0              Se posiciona el cursor en la posición 5.0 del LCD
PRINT FORMAT(a,DEC,2)   El valor de entrada, a, se envía al módulo LCD con formato decimal de dos dígitos.
GOTO MAIN               Salta a MAIN.
```

(4) Configuraciones del giróscopo

- Los giróscopos se pueden conectar a los puertos AD 0,1,2,3,4,5,6,7.
- Si se usan giróscopos GWS-PG03 (GWS), puede usar como 4 de ellos a la vez.
- Si usa un giróscopo KRG-1 (Kondo), debe usarlo en modo normal para los dos modos.
(Modo Robot/Valor por defecto. Modo normal). Para más información, consulte al fabricante del giróscopo.

La función Gyro se explica en el "Apartado 11, explicación de comandos RoboBasic para el proceso de señales analógicas" del manual incluido en el CD.

GYRODIR

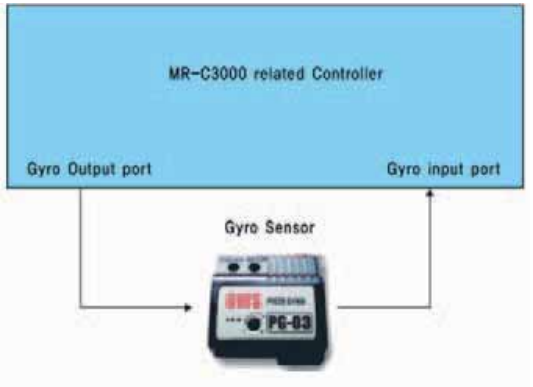
Define el sentido de giro de los servos si se usa un giróscopo.

Sintaxis

GYRODIR [Grupo], [Sentido]

Explicación del comando

Este proceso controla el sentido de giro de un grupo de servos cuando se conecta un giróscopo al puerto AD de un controlador de la serie MR-C3000. El número máximo de giróscopos que se pueden usar es 4. Vea la siguiente tabla.



AD Port Number (Digital In and Out Port Number) of MR-C3000 series controller.	Gyro Port
Port #0 (Port #32)	Gyro #1 channel output Port
Port #1 (Port #33)	Gyro #2 channel output Port
Port #2 (Port #34)	Gyro #3 channel output Port
Port #3 (Port #35)	Gyro #4 channel output Port
Port #4 (Port #36)	Gyro #1 channel input Port
Port #5 (Port #37)	Gyro #2 channel input Port
Port #6 (Port #38)	Gyro #3 channel input Port
Port #7 (Port #39)	Gyro #4 channel input Port

Dado que un giróscopo es reversible, se debe especificar la dirección del giro en el siguiente parámetro. [Sentido] debe ser "0" o "1". Si vale 1, se incrementa la posición del servo, 0 la disminuye.

Ejemplo del comando

GYRODIR G6A, 1, 1, 0, 0, 1, 0

GYROSET

Define que giróscopo controlará un grupo determinado de servos.

Sintaxis

GYROSET [Grupo], [N Gyro] …

Explicación del comando

Gyroset determina cual de los servos en un [grupo] se controla por un giróscopo determinado.
[N Gyro] es el puerto del giróscopo que se usa para cada servo del grupo. Vea el ejemplo inferior.

Ejemplo del comando

GYROSET G6B, 1, 1, 2, 2, 0, 0
El servo #6 recibe el sensor del giróscopo # r1 y lo procesa
El servo #7 recibe el sensor del giróscopo # r1 y lo procesa
El servo #8 recibe el sensor del giróscopo # r2 y lo procesa
El servo #9 recibe el sensor del giróscopo # r2 y lo procesa
El servos #10 y #11 no usan el sensor del giróscopo .

01,02,03,04 : GWS PG03 11,12,13,14 : KRG-1
21,22,23,24 : reservado 31,32,33,34 : reservado

GYROSENSE

Define la sensibilidad del servo a un giróscopo.

Sintaxis

GYROSENSE [Grupo], [sensibilidad N Gyro]

Explicación del comando

Se pueden conectar 4 giróscopos a un controlador de la serie MR-C3000. GYROSENSE define la sensibilidad de un servo individual a un giróscopo.
[sensibilidad N Gyro] usa números entre 0 y 255 o constantes para controlar la sensibilidad de cada servo en un grupo. Un valor de "0" no cambia la sensibilidad de los servos. A medida que el número aumenta, la reacción del servo respecto al giro aumenta también

Ejemplo del comando

GYROSENSE G6A, 100, 100, 255, 255, 50, 50
Servos #0 y #1 se configuran con sensibilidad 100 del Gyro.
Servos #2 y #3 configurados con la sensibilidad máxima (255) del gyro.
Servos #4 y #5 se configuran con sensibilidad 50 del giro.

1) Como conectar giróscopos GWS, KRG-1
* Puerto del giróscopo #1: ADO, AD4
* Puerto del giróscopo #2: AD1, AD5
* Puerto del giróscopo #3: AD2, AD6
* Puerto del giróscopo #: AD3, AD7

Ex:
Los servos #1, #2, #3 del grupo A y los servos #1, #2, #3 del grupo D usan el giróscopo #1
Los servos #1, #2, #3 del grupo A y los servos #1, #2, #3 del grupo D usan el sentido de giro 0
Los servos #1, #2, #3 del grupo A y los servos #1, #2, #3 del grupo D usan las ganancias 250,200,250

(Código para un giróscopo GWS)

GYROSET G6A,0,1,1,1,0,0
GYROSET G6D,0,1,1,1,0,0

GYROSET G6A,0,0,0,0,0,0
GYROSET G6D,0,0,0,0,0,0

GYROSENSE G6A, 0, 250, 200, 250, 0, 0
GYROSENSE G6D, 0, 250, 200, 250, 0, 0

(Código para un giróscopo KRG-1)

GYROSET G8A,11,10,10,10,10,10,10,10
GYROSET G8B,10,10,10,10,10,10,10,10
GYROSET G8C,10,10,10,10,10,10,10,10
GYROSET G8D,10,10,10,10,10,10,10,10
'.....
GYROSENSE G8A,255,1,1,1,1,1,1,1
GYROSENSE G8B,1,1,1,1,1,1,1,1
GYROSENSE G8C,1,1,1,1,1,1,1,1
GYROSENSE G8D,1,1,1,1,1,1,1,1

GYRODIR G8A,1,1,1,1,1,1,1,1
GYRODIR G8B,1,1,1,1,1,1,1,1
GYRODIR G8C,1,1,1,1,1,1,1,1
GYRODIR G8D,1,1,1,1,1,1,1,1

(5) Como usar I2C

Es un tipo de interface para comunicaciones.
Puede usarse si se dispone de 2 puertos para recibir y transmitir una señal.

```
CONST scl  =22
CONST sda  =23

dim A as byte

'========================
I2C_SAMPLE:
'========================
 a = IN(sda)
s1: OUT scl,0
 OUT scl,1
'========================
s3: OUT sda,0
 OUT scl,0
 OUT scl,1
'========================
 OUT sda,1
 OUT scl,0
 OUT scl,1

 OUT sda,0
 OUT scl,0
 OUT scl,1

 OUT sda,0
 OUT scl,0
 OUT scl,1

 OUT sda,1
 OUT scl,0
 OUT scl,1
'========================
 CODE = 0
 FOR  I =  0 TO 7
 OUT SCL,0
 OUT SCL,1
 A = IN(SDA)
 A = A<<I
 CODE = CODE OR A
 NEXT I
'========================
 RETURN
'========================
```

6) Pulsos del protocolo HMI

HMI (Hitec Multi-protocol Interface) es un nuevo tipo de interface diseñado para los servos programables de robots HITEC. (Para más información sobre la programación de los servos, hace falta un kit de configuración opcional.)

* La siguiente información sólo es par usuarios avanzados, ya que implica la transmisión de pulsos. Estos pulsos (datos) son necesarios para controlar la posición de los servos de robots HITEC y comunicarlas al PC.

Los servos ejecutarán un movimiento determinado, dependiendo de los cuatro tipos de pulsos recibidos desde el exterior.

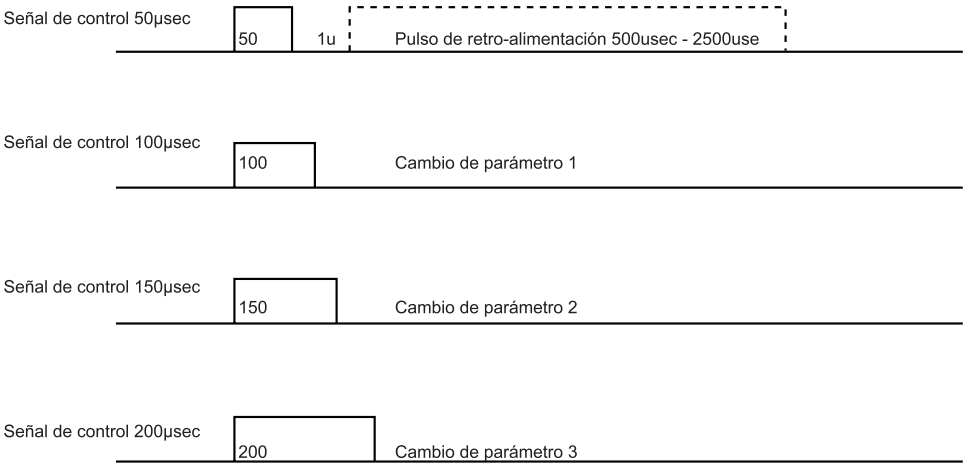
1) 50usec de amplitud de pulso / Retro-alimentación de posición (ángulo)

2) 100usec de amplitud de pulso / Usa el valor del parámetro 1 del servo (default)

3) 150usec de amplitud de pulso / Usa el valor del parámetro 2 del servo

4) 200usec de amplitud de pulso / Usa el valor del parámetro 2del servo

Nota: para usar la retro-alimentación con un circuito externo, los terminales de control de señales deben ser bidireccionales. Para recibir la retro-alimentación (datos), debe realizar un proceso pull-up en la línea de señal.



Nota: La función de retro-alimentación tiene una tolerancia del 10% ya que esta función se usa con el control de PWM.

ROBONOVA-I

(7) Configuración del aviso por voltaje bajo (LOW-VOLTAGE WARNING)

Cuando el voltaje de la batería del ROBONOVA-I cae por debajo de un punto, el LED del ROBONOVA-I instalado en la cabeza parpadeará para indicar al usuario que es imprescindible que recárguelas baterías inmediatamente.

El valor del voltaje mínimo por defecto es de 5.8V. puede cambiar mediante programación.

```
0029 '================================================
0030 MAIN:
0031 'GOSUB robot_voltage
0032 GOSUB tilt_proc
0033
0034 IF RR > 50 THEN GOTO action_proc
0035 '------------------------------
0036 IF RR = 0 THEN GOTO MAIN1
0037
0038 ON RR GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K
0039 GOTO main_exit
0040 '------------------------------
```

Quite la marca de comentario (') desde 'GOSUB robotvoltage que aparecen en la rutina MAIN
Ex: Cambie 'GOSUB robot_voltage por GOSUB robot_voltage

```
'================================================
robot_voltage:                          ' [ 10 x Value / 256 = Voltage]
    DIM v AS BYTE

    A = AD(6)

    IF A < 148 THEN                     ' 5.8v

    FOR v = 0 TO 2
    OUT 52,1
    DELAY 200
    OUT 52,0
    DELAY 200
    NEXT v

    RETURN
```

Salta a la rutina robot_voltage.

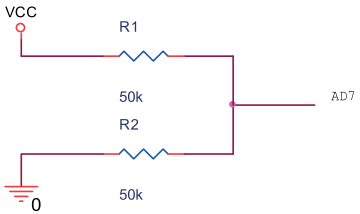
Para cambiar el valor del voltaje mínimo, inserte las siguientes instrucciones;

IF A < [valor] THEN
(En este ejemplo, valor = voltaje*256/10)

Ex: Si el voltaje mínimo ahora es 6.0v, el [valor] será 153.6 (6.0 *256/10 = 153.6). Ya que RoboBasic no usa decimales, se usarán el valor 153.

IF A< 153 THEN
6.0 *256/10 = 153.6 = 153 -> IF A< 153 THEN

3. Circuito de detección de bajo voltaje



ROBOBASIC

Manual de comandos

v2.10

Marca registrada

Windows es una marca registrada de Microsoft Corporation. ROBOBASIC es software registrado por miniROBOT, INC.

Nota

Este manual explica los comandos utilizados en roboBASIC. Hitec no se hace responsable por los usos erróneos que se puedan hacer. Este manual puede ser modificado sin previo aviso para mejorar las prestaciones del producto.

RoboBasic es un software registrado, por lo tanto es ilegal el reproducir, publicar, difundir públicamente, transmitir o distribuir este manual o el software sin permiso explicito.

- Índice -

Capítulo 1	Sumario de comandos de ROBOBASIC	2
Capítulo 2	Gramática general de ROBOBASIC	5
Capítulo 3	Explicación de los comandos de declaración en ROBOBASIC	9
Capítulo 4	Explicación de los comandos de control de flujo en ROBOBASIC	10
Capítulo 5	Explicación de las señales de entrada y salida en ROBOBASIC	16
Capítulo 6	Explicación de los comandos relacionados con la memoria	20
Capítulo 7	Detalles del módulo LCD en ROBOBASIC	21
Capítulo 8	Detalles de los comandos para el control de motor en ROBOBASIC	25
Capítulo 9	Comandos para el control de música	36
Capítulo 10	Comandos para comunicaciones externas	39
Capítulo 11	Descripción de los comandos para el proceso de señales analógicas	43
Capítulo 12	Comandos de proceso y otros	48
Capítulo 13	Descripción de comandos	49

Capítulo 1

Sumario de comandos

para roboBASIC

Sumario de comandos

RoboBASIC es un exclusivo lenguaje de programación diseñado para controlar robots. En roboBASIC, los comandos necesarios para controlar un robot han sido añadidos al lenguaje BASIC.

└2)	Este símbolo indica que el comando solo se puede ejecutar en controladores de la serie MR-C2000.
└3)	Este símbolo indica que el comando solo puede ejecutar en controladores de la serie MR-C3000.

Comandos usados para declaraciones/definiciones

	DIM	Declara una variable
	AS	Asigna una variable durante su declaración
	CONST	Declara una constante
	BYTE	Se define una variable como tipo byte en su declaración
	INTEGER	Se define una variable como tipo integer (entero) en su declaración

Comandos para el control de flujo

	IF	Comienzo de una sentencia condicional
	THEN	Se ejecuta la siguiente sentencia (o sentencias) cuando la condición es cierta
	ELSE	Se ejecuta la siguiente sentencia (o sentencias) cuando la condición es falsa.
	ELSEIF	Comienzo de otra sentencia condicional.
	ENDIF	Fin de una sentencia condicional
	FOR	Comienza un bucle.
	TO	Asigna el rango de iteraciones de un bucle. NEXT Sentencia de cierre de ejecución de bucle
	GOTO	Divide el flujo del programa.
	GOSUB	Invoca a una sub-rutina
	RETURN	Sale (retorna) de una sub-rutina
	END	Finaliza la ejecución del programa. STOP Detiene la ejecución del programa .
	RUN	Ejecuta un programa (continua).
	WAIT	Espera hasta que se complete el programa.
	DELAY	Hace una pausa en el programa durante un tiempo.
(2)	BREAK	Detiene la ejecución del programa y pasa a modo de depuración.

Comandos de entrada y salida de señales digitales

	IN	Lee las señales de un puerto de entrada.
	OUT	Envía una señal a un puerto de salida
	BYTEIN	Lee una señal byte de un puerto de entrada
	BYTEOUT	Envía una señal byte a un puerto de salida.
(2)	INKEY	Tecla (clave) proveniente de un puerto de entrada.
	STATE	Estado del puerto de salida
	PULSE	Envía una señal (de pulso) al puerto de salida.
	TOGGLE	Invierte el estado del puerto de salida.
(3)	KEYIN	Toma la entrada del teclado analógico (de entrada).

Comandos de memoria

	PEEK	Lee datos del controlador de la RAM.
	POKE	Escribe datos en el controlador de la RAM.

	ROMPEEK	Lee datos del controlador externo EEPROM RAM.
	ROMPOKE	Escribe datos en el controlador EEPROM RAM.

Comandos para el LCD

	LCDINIT	Inicializa el módulo LCD.
	CLS	Borra todos los caracteres del módulo LCD.
	LOCATE	Define la posición de caracteres en el módulo LCD.
	PRINT	Muestra caracteres en el módulo LCD.
	FORMAT	Define el formato a mostrar en el módulo LCD
	CSON	Hace que aparezca el cursor en el módulo LCD.
	CSOFF	Oculta el cursor del módulo LCD.
	CONT	Define el contraste de los caracteres en el módulo LCD.
	DEC	Envía un valor decimal al módulo LCD.
	HEX	Envía un valor hexadecimal al módulo LCD.
(3)	BIN	Envía un valor binario al módulo LCD

Operadores lógicos

	AND	Usa la operación lógica AND.
	OR	Usa la operación lógica OR.
	MOD	Calcula el módulo de una operación aritmética.
	XOR	Usa la operación lógica XOR.
(3)	NOT	Invierte todos los bits

Comandos para el control del motor

	ZERO	Define el punto 0 (neutro) del servo.
	MOTOR	Activa el puerto de salida de un servo.
	MOTOROFF	Desactiva el puerto de salida de un servo.
	MOVE	Maneja varios servos al mismo tiempo.
	SPEED	Define la velocidad del servo.
(2)	ACCEL	Define la aceleración del servo.
	DIR	Define la dirección (sentido) del motor del servo.
	PTP	Activa/desactiva varias operaciones del control del motor.
	SERVO	Controla el servo.
	PWM	Define la amplitud del pulso de un motor DC.
(2)	FASTSERVO	Maneja el servo a la velocidad máxima.
(3)	HIGHSPEED	Activa/desactiva la velocidad máxima del servo.
(3)	MOVEPOS	Mueve el grupo de servos definidos con POS.
(3)	POS	Selecciona una posición especifica del robot.
(3)	FPWM	Cambia la amplitud y la frecuencia del pulso
(3)	MOVE24	Maneja los 24 servos a la vez.
(3)	INIT	Define la posición inicial
(3)	MOTORIN	Lee la posición actual del servo (valor).
(3)	AIMOTOR	Configuración de uso del motor AI.
(3)	AIMOTOROFF	Cancela el motor AI.
(3)	AIMOTORIN	Lee la posición actual del motor AI (valor).
(3)	SETON	Configuración para usar la función “setup ”
(3)	SETOFF	Cancela la función “set up”.
(3)	ALLON	Función de configuración para todos los servos.
(3)	ALLOFF	Cancela la función de config. de todos los servos.
(3)	GETMOTORSET	Lee los valores de un servo y mantiene la posición.

Parámetros para asignación de grupos de motores

(3)	G6A	Asigna los servos #0~#5 al grupo A.
(3)	G6B	Asigna los servos #6~#11 al grupo B.
(3)	G6C	Asigna los servos #12~#17 al grupo C.
(3)	G6D	Asigna los servos #18~#23 al grupo D.
(3)	G6E	Asigna los servos #24~#29 al grupo E.
(3)	G8B	Asigna los servos #8~#15 al grupo B.
(3)	G8C	Asigna los servos #16~#23 al grupo C.
(3)	G8D	Asigna los servos #24~#31 al grupo D.
(3)	G12	Asigna los servos #0~#11.
(3)	G16	Asigna los servos #0~#15.
(3)	G24	Asigna los servos #0~#23.
(3)	G32	Asigna los servos #0~#31.

Comandos para el control de sonidos

(2)	BEEP	Emite una alerta con el PIEZO.
(2)	SOUND	Reproduce una frecuencia con el PIEZO.
(2)	PLAY	Reproduce un sonido con el PIEZO.
(3)	MUSIC	Reproduce música con el PIEZO.
(3)	TEMPO	Define el ritmo de un sonido.

Comandos para comunicaciones con el exterior

(2)	RX	Recibe una señal RS-232 por el puerto RX.
(2)	TX	Envía una señal RS-232 por el puerto TX.
(2)	MINIIN	Recibe una señal minibus por el puerto de comunicaciones mini.
(2)	MINIOUT	Envía una señal minibus por el puerto de comunicaciones mini.
(3)	ERX	Recibe una señal RS-232 por el puerto RX.
(3)	ETX	Envía una señal RS-232 por el puerto TX.

Comandos para el proceso de señales analógicas

(3)	AD	Lee señales analógicas por el puerto AD.
(3)	REMOCON	Lee el valor de una tecla por el control remoto infrarrojos.
(3)	SONAR	Lee la distancia desde el puerto del sensor de ultrasonidos
	RCIN	Lee el valor de entrada del controlador remoto RC.
(3)	GYRODIR	Define la dirección del giróscopo.
(3)	GYROSET	Asigna un giróscopo a un servo.
(3)	GYROSENSE	Define la sensibilidad (ganancia) de un giróscopo.

Procesado de comandos

ON...GOTO Salto dependiendo del valor de una variable.

Otros comandos

	RND	Crea un número aleatorio.
	REMARK	Comentario

Comandos de intención

	$DEVICE	Configura el controlador para ser manejado por el programa en ejecución.
(3)	$LIMIT	Limita el recorrido de un servo.

Capítulo 2

Gramática general

de roboBASIC

Ya que la gramática de roboBASIC se basa en el lenguaje de programación BASIC genérico, la mayoría del roboBASIC es muy parecido, o igual, que el BASIC. En este capítulo, explicaremos la gramática general de roboBASIC.

Juego de caracteres

El juego de caracteres de roboBASIC se compone de las letras inglesas (A-Z, a-z), números (0-9) y símbolos especiales. Los símbolos mostrados en la siguiente tabla tienen un significado especial en roboBASIC.

Símbolo	Descripción
+	Símbolo suma
-	Símbolo resta
*	Símbolo de multiplicación
/	Símbolo de división
%	Símbolo remanente (porcentaje)
.	Símbolo de designación de bit
&	Símbolo numérico
??	Símbolo de texto
??	Símbolo de cadena de caracteres (string)
:	Símbolo de etiqueta
=	Símbolo igual o símbolo de sustitución
<	Símbolo de diferencia (no igual)
>	Símbolo de diferencia (no igual)
<<	Desplazamiento de bit a la izquierda
>>	Desplazamiento de bit a la derecha

Formulas y operadores

Las formulas pueden estar compuestas por valores, resultante de otras fórmulas, variables, y números relacionados entre ellos mediante operadores. Un operador ejecuta un cálculo numérico o lógico partiendo de un valor determinado. En roboBASIC, los operadores se pueden clasificar como se muestra en la tabla inferior.

Clasificación	Función
Operador aritmético	Lleva a cabo un cálculo aritmético.
Operador relacional	Compara valores numéricos.
Operador lógico	Compara condiciones o manipula bits
operador de bit	Manipula bits o ejecuta una operación con bits.

Operadores aritméticos

Un operador aritmético es un símbolo que realiza un cálculo. Al igual que en el lenguaje genérico BASIC, la suma (+), la resta (-), la multiplicación (*), la división (/), y el módulo (% o MOD) se pueden usar en roboBASIC. Sin embargo, existen algunas diferencias entre roboBASIC y el BASIC genérico.

En primer lugar, no hay prioridades entre operadores

En roboBASIC, no se pueden usar los paréntesis ( ).

Ejemplo: A = 1, B = 2, C = 3
Basic genérico: A + B * C = 1 + 2 * 3 = 1 + 6 = 7 (en BASIC, se podrían usar los paréntesis si la suma tuviese

prioridad sobre la multiplicación)

RoboBASIC: A + B * C = 1 + 2 * 3 = 3 * 3 = 9

En segundo lugar, los cálculos matemáticos complejos pueden causar errores inesperados.

Si fuese así, el cálculo debe dividirse en varios más sencillos.

Ejemplo:

D = A * B + C (Se acepta)
F = A * B / C * D + E (Evite cálculos complejos como este)

Tercero, roboBASIC solo soporta tipos byte o integer, de manera que los puntos decimales se ignoran.

Los cálculos del módulo usan el símbolo "%" o "MOD" y su resultado será un módulo.

Operadores relacionales

Un operador relacional se usa para comparar dos valores. El resultado será "TRUE" o "FALSE". El resultado se suele utilizar para controlar el flujo del programa mediante una sentencia IF.

Operador	Relación	expresión
=	Igual que	X = Y
<>	Distinto de	X <> Y
<	Menor que	X < Y
>	mayor que	X > Y
<=	Menor o igual que	X <= Y
>=	Mayor o igual que	X >= Y

Cuando se combinan en una formula un operador aritmético con otro lógico, el operador aritmético se ejecutará antes que el lógico.

Operadores lógicos

Un operador lógico se usa para comparar condiciones combinadas. El resultado siempre devolverá "TRUE" o "FALSE". El valor devuelto se usa para controlar el flujo del programa en una sentencia IF.

Operador	Significado
AND	Y
OR	O
XOR	O exclusivo

Cada operador devuelve un valor como los indicados en la tabla inferior. En la tabla, "T" significa verdadero("true"), "F" significa falso ("false").

Valor de X, Y		Resultado		
X	**Y**	**X AND Y**	**X OR Y**	**X XOR Y**
T	T	T	T	F
T	F	F	T	T
F	T	F	T	T
F	F	F	F	F

Operadores binarios

Un operador binario realiza cálculos de cada variable que se use en el controlador del robot haciendo más sencillo el control de los bits a través del puerto I/O (input/output).
Existe la suma de bits (OR), multiplicación (AND) y la suma exclusiva (XOR) del conjunto de bits. En roboBASIC, los símbolos de cálculos, izquierda (<<), derecha (>>) y ".", se usan para mover un bit a una posición determinada (desplazamiento).

Si el valor de A es 33 (en binario 00100001) y el valor de B es 15 (en binario 00001111), al usar los operadores comentados se devolverían los siguientes resultados.

operador	resultado
A AND B	1 (00000001)
A OR B	47 (00101111)
A XOR B	46 (00101110)
A << 1	66 (01000010)
A >> 1	16 (00010000)
A.0	1 (0 Bit de A)

Si se usan varios operadores en la misma instrucción, las operaciones se llevarán a cabo en el siguiente orden:

① Operador aritmético/ Operador binario

② Operador relacional

③ Operador lógico

Tipos, Variables/Constantes y otras consideraciones gramaticales

Ya que roboBASIC está diseñado para el control de robots, roboBASIC no soporta variables ni constantes relacionadas con cadenas de texto (strings), al contrario de lo que ocurre en BASIC.

Tipos (números)

Existen tipos byte y tipos integer. El rango de los posibles valores para cada uno de los tipos se muestra en la tabla siguiente.

Tipo de dato	Tamaño	Rango
BYTE	1 byte (8bit)	0-255
INTEGER	2 byte (16bit)	0-65535

RoboBASIC no soporta valores negativos. Si se añade un signo "+" o "–" precediendo a un número, la operación causará un error.
Las declaraciones deben hacerse teniendo en cuenta los valores de cada tipo.

Antilogaritmos

Ya que roboBASIC está diseñado para controlar hardware, la utilización de números hexadecimales es mucho más eficiente que el uso de números decimales. En roboBASIC, puede usar número binarios (Bin), octales (Oct), decimales (Dec), y hexadecimales (Hex).

Antilogaritmo	Declaración	Valor permitido	Ejemplo
Número binario	&B	0, 1	&B111101
Número octal	&O	0, … , 7	&O75
Número decimal	N/A	0, … , 9	61
Número hexadecimal	&H	0, … , 9, A, … F	&H3D

Constantes y variables

El valor de una constante no cambia durante la ejecución del programa. RoboBASIC puede definir una constante como de tipo byte o de tipo integer. El tipo de la constante se define automáticamente dependiendo del valor asignado. Una vez definida la constante, no se puede redefinir. La definición de una constante no tiene implicaciones en el tamaño del programa. Siempre que vaya a usar un número de manera habitual en el programa, deberá definirlo como constante.

Ejemplo

```
CONST OFF = 0
CONST motor_1 = 3
CONST motor_1 speed = 200
```

Una variable es el nombre de una posición de memoria donde se almacenarán valores durante la ejecución del programa. En el controlador minirobot, el número de variables es limitado, de manera que la definición de las variables debe hacerse para minimizar el consumo de memoria y de acuerdo al tipo de valor a almacenar.

```
DIM motor_1_delay AS INTEGER
DIM sensor_left AS BYTE
```

Cuando declare una variable o constante, siga las siguientes normas.

Primero: Puede usar inglés o coreano en el primer carácter. En coreano (Chino) o inglés, puede usar caracteres alfanuméricos o el símbolo "_" en el nombre de una variable o constante.

Segundo: La longitud del nombre no puede sobrepasar los 64 caracteres.

Tercero: La variable o la constante no puede definirse dos veces con el mismo nombre y no hay distinción entre mayúsculas y minúsculas.

Cuarto: Si declara una constante con un valor superior a 65535, que es el valor máximo de un entero (integer), puede producirse un error.

Puntuación en Bits

En roboBASIC, las variables se pueden manejar como unidades de bits. Para hacerlo así, se usa el operador de puntuación de bits (".”). Cuando use el operador de bits, los bits 0~7 (variable de tipo byte) y los bits 0~15 (variable de tipo integer) serán validos. Sólo se pueden usar números y constantes.

Ejemplo

```
DIM A AS INTEGER
CONST BIT_2 = 2

A.1 = 1
A.BIT_2 = 0
A.3 = IN(1)    'Lee un valor del puerto #1 y lo asigna al tercer bit de la variable A

OUT 2, A.1     'Envía el valor del 1er bit de la variable A al puerto #2
```

Explicación de las sentencias
Los comentarios explicando el programa deben intercalarse en su interior para facilitar su comprensión y diseño. Para insertar un comentario (explicación), puede usar el símbolo (') o la instrucción "REMARK". La introducción de este tipo de sentencias (comentarios) en el programa no afecta la ejecución de éste.

Sentencia de substitución (=)
Una sentencia de substitución se usa para sustituir un valor en una variable. Se usa el símbolo "=". El valor siempre se coloca a la izquierda del signo (=) y la variable, cadena de caracteres, fórmula o función se pone a la derecha del signo (=).

Ejemplo

```
A = B          'Substituye cada variable
```

```
A.1 = 1          'Substituye el bit indicado
A = ADIN(0)  'Función de substitución.
A = 3 * 2 - 1    'Substituye el cálculo numérico de la formula
A = C + B - A 'Substituye la variable calculada en la formula
A = "1"          'Substituye el código ASCII
```

Etiqueta de línea
La etiquetas de línea se usan para "apuntar" a una ubicación concreta dentro del programa. Para estas etiquetas puede usar caracteres y números. Hay algunas reglas para crear etiquetas.

Primero: Una etiqueta no debe exceder de 64 caracteres y el primero debe ser un carácter inglés o coreano.

Segundo: El símbolo de etiqueta (:) debe seguir a la etiqueta.

Tercero: Los números incluidos en el rango 0 ~ 65535 se pueden usar en el nombre de la etiqueta. En este caso no es necesario el símbolo.

Cuarto: El nombre de la etiqueta no puede estar duplicado y no se distingue entre mayúsculas y minúsculas. Normalmente, las etiquetas se suelen usar en comandos de control de flujo como GOTO o GOSUB.

Ejemplo

```
DIM A AS INTEGER
START:
        A = IN(0)
        IF A = 0 THEN
                GOTO START
        ELSE
                GOSUB 10
        END
        GOTO START
10 OUT 1, 0
        DELAY 100
        OUT 1, 1
        RETURN
```

Capítulo 3

Explicación de los comandos

de declaración en roboBASIC

Estos comandos se usan para la declaración de variables y constantes.

DIM ... AS **Declare ...as**
Declara una variable

Sintaxis
① Para declarar una única variable :

- Sentencia: DIM [nombre de la variable] AS [tipo de la variable]

② Para declarar múltiples variables:

- Sentencia: DIM [nombre de la variable] AS [tipo], [nombre…] AS [tipo]…

Explicación del comando
Una variable usada en roboBASIC debe declararse con el comando DIM. El comando DIM debe usar la opción AS para declarar el tipo de la variable. El nombre de la variable no distingue entre mayúsculas y minúsculas. El nombre de la variable debe ser único. Una variable se usa para procesar el valor de un sensor o el valor de una señal analógica una vez convertido. Usando las variables apropiadas, la creación del programa será más eficiente. El número de las variables a usar es distinto para cada controlador de robots.
La serie MR-C2000 usa variables con un tamaño inferior a 30. La serie MR-C3000 usa variables con un tamaño inferior a 256 bytes. Las variables de tipo Byte ocupan 1 byte y las e tipo integer ocupan 2 bytes de manera que la declaración debe ser la apropiada para
No exceder el número máximo de variables.

Ejemplo del comando

```
DIM I AS INTEGER      'Declara I como tipo integer
DIM J AS BYTE         'Declara J como tipo byte
```

CONST
Declara una constante

Sintaxis
Sentencia: CONST [nombre de la constante] = Valor

Explicación del comando
Nombrar a un valor mediante una constante simplifica el programa. Algunas de las ventajas de su uso, respecto a valores y variables son:
① Una vez definida, puede usarse en todo el programa.
② Las constantes no se pueden modificar por error.
③ Su modificación es muy sencilla.
④ Una constante ocupa poca memoria.

Ejemplo del comando

```
CONST OFF = 0         'Declara OFF como 0 (constante)
CONST A = &HB1001     'Declare A con número decimal 9 (constante)
```

Capítulo 4

Explicación de los

comandos de control de flujo

Estos comandos se usan para controlar el flujo de ejecución del programa.

IF ... THEN ...
Sentencia condicional

Sintaxis
① Condición única:
- sintaxis: IF [condición] THEN [sentencias a ejecutar cuando la condición se cumple]

② Múltiples condiciones:
- sintaxis: IF [condición1] THEN [sentencias a ejecutar si se cumple la condición1]
ELSEIF [condición 2] THEN [sentencias a ejecutar si se cumple la condición2]

```
ELSE [sentencias a ejecutar si no se cumplen las condiciones anteriores]
ENDIF
```

Explicación del comando

Cuando se ejecuta una sentencia IF ... THEN, se evalúa la condición que sigue a IF. Si se cumple la condición (TRUE), se procesa la sentencia THEN. Si no se cumple la condición, (FALSE), se evaluarán las condiciones ELSEIF y se ejecutará la apropiada. Si no se cumple ninguna, se ejecutará la sentencia ELSE. Puede usar también la sentencia, ELSEIF, si fuese necesario.
En roboBASIC, la sentencia (IF..THEN) es crucial para trabajar con señales externas de entrada y almacenar su valor en una variable. La sentencia condicional evalúa una variable y permite al robot moverse de acuerdo con este valor obtenido y procesado.

Ejemplo del comando

① La ejecución de la condición y la sentencia es muy sencilla. Ambas pueden incluirse en la misma línea.

```
IF A > 0 THEN B = 5
IF A < 5 THEN B = 0 ELSE B = 1
```

② La condición a evaluar de una sentencia IF puede usar dos tipos de condiciones si se usa un operador relacional.

```
IF A > 0 AND A < 5 THEN B = 3
IF A = 7 OR A = 9 THEN B = 1
```

③ Ejemplo de una sentencia IF un poco más compleja

```
IF A = 1 THEN
        B = 2
        C = 3
ELSEIF A = 3 AND A = 5 THEN
        B = 1
        C = 2
ELSEIF A = 8 THEN
        B = 6
        C = 0
ELSE
        B = 0
        C = 0
ENDIF
```

FOR ... NEXT

Bucle (Repetición de sentencias un numero determinado de veces)

Sintaxis

- Sintaxis:

```
FOR [contador] = [inicio] TO [final]
        [sentencias a ejecutar]
NEXT [contador]
```

Explicación del comando

La variable [Contador] cuenta el número de repeticiones. [Inicio] es el valor inicial de la variable contador, [final] es el valor máximo del contador. Para los valores de [Inicio] y [Final] se puede usar un valor, una constante o una variable.
En roboBASIC, el valor de [Final] debe ser mayor que el valor de [Inicio]. RoboBASIC incrementa el valor de la variable contador. Existen algunas reglas a la hora de usar las sentencias FOR...NEXT.

① Una bucle FOR...NEXT puede usarse dentro de otro FOR...NEXT (Anidamiento).

```
FOR I = 1 TO 10
        FOR J = 1 TO 5
                .....
        NEXT J
NEXT I
```

② Al usar varias sentencias FOR...NEXT, el orden de NEXT [contador] no puede cambiarse.

Incorrecto

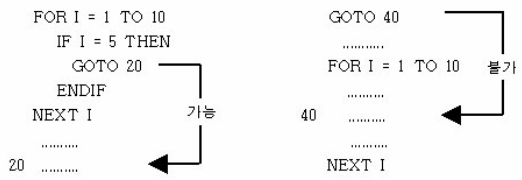
```
FOR I = 1 TO 10
        FOR J = 1 TO 5
                .....
        NEXT I
NEXT J
```

Correcto

```
For I = 1 to 10
        For J = 1 to 5
                ……
        Next J
Next I
```

Aunque hubiésemos usado el ejemplo anterior marcado como incorrecto, no se producirá ningún error hasta que hayamos volcado el código al robot.

③ Es posible salir del interior de un bucle FOR…NEXT pero es completamente imposible entrar directamente a una de sus líneas desde el exterior



④ El valor de las variables [Contador], [Inicio] y [Final] no se deberían cambiar arbitrariamente dentro de un bucle FOR...NEXT.

Ejemplo del comando

Conecte el L.E.D. al puerto #0 del controlador y haga que destelle 5 veces.

```
DIM A AS BYTE     'Declara la variable a usar como contador
FOR A = 1 TO 5    'Se repite 5 veces
OUT  0,  0        'Enciende el L.E.D. conectado al puerto #0
        DELAY 1  0  'Pausa de 100
OUT 0, 1          'Apaga el L.E.D. del puerto #0
        DELAY 100   'Pausa de 100
NEXT A
```

GOTO

Salta a un punto especificado.

Sintaxis

- Sintaxis: GOTO [Etiqueta]

Explicación del comando

El comando GOTO cambia el flujo de ejecución del programa saltando a una línea específica. La utilización del comando GOTO complica demasiado la comprensión de la ejecución del programa. No lo use demasiado.

Ejemplo

```
DIM I AS INTEGER
DIM J AS BYTE

I = 7
```

```
        IF I = 6 THEN GOTO L1
            .....
L1:     J = 1
        OUT I, J
```

GOSUB ... RETURN

Llama a una sub-rutina (invoca) y vuelve.

Sintaxis

```
- Sintaxis:     GOSUB [Etiqueta]
               .....
[Etiqueta]:    .....
               RETURN
```

Explicación del comando

El comando GOSUB hace una llamada a una rutina usada frecuentemente y vuelve a la siguiente línea. De esta manera, GOSUB hace que el programa sea más pequeño y eficiente.

Es posible llamar a otra subrutina desde dentro de la que se invocó originalmente. Esto está limitado a 4 veces en la serie MR-C2000 y 5 veces en la serie MR-C3000. Hacerlo más veces, producirá un error.

Ejemplo del comando

```
        DIM LED_PORT AS INTEGERL
        LED_PORT = 1
        START: .....
        .....
        GOSUB LED_TOGGLE
        .....
        GOTO START END
LED_TOGGLE:TOGGLE LED_PORT
        RETURN
```

END

Finaliza la ejecución del programa.

Sintaxis

- Sintaxis : END

Explicación del comando

Al pasar 2 segundos desde que se conecta el controlador del robot, el programa almacenado en al EEPROM será procesado (ejecutado). Si no se usa el comando END al final de una rutina o como última línea del código, el programa seguirá ejecutándose indefinidamente. Incluya siempre la instrucción END para terminar una rutina o el código.

Ejemplo del comando

① Finaliza la ejecución de un programa tras completarse.

```
        DIM A AS BYTE
        START: A = IN(0)
                IF A = 1 THEN END
                .....
                GOTO START
```

② Puede usarlo para finalizar una estructura de sub-rutinas.

```
DIM A AS BYTE
```



```
A = BYTEIN(0)
IF A = 1 THEN
   GOSUB L1
ELSEIF A = 3 THEN
   GOSUB L2
ELSEIF A = 4 THEN
   GOSUB L3
ELSE
   GOSUB L4
ENDIF
..... END

L1: .....
        RETURN
L2: .....
        RETURN
L3: .....
        RETURN
L4: .....
        RETURN
```

STOP/RUN

Detiene/Comienza la ejecución de un programa

Sintaxis

-Sintaxis: STOP/RUN

Explicación del comando

Este comando le permite detener y reiniciar un programa. Cuando el programa se haya detenido, puede usar el comando RUN para comenzar de nuevo.

WAIT

Espera hasta que un programa finaliza.

Sintaxis

- Sintaxis: WAIT

Explicación del comando

El sistema operativo instalado en el controlador del robot procesa en TIEMPO-REAL la ejecución de los programas. Cuando se ejecuta un programa, otro programa puede ejecutarse al mismo tiempo sin detener la ejecución del anterior. Si fuese necesario que un programa se ejecute solo cuando el anterior haya terminado, debe usar el comando WAIT.

Ejemplo del comando

Ex 1: Puertos de salda #7 y #8 después de mover seis motores.

```
MOVE 120, 100, 140, 90, 70, 150
WAIT
OUT 7, 1
OUT 8, 1
```

Ex 2: Escribe en #8, tras manejar los seis motores y enviar al #7 al mismo tiempo.

```
MOVE 120, 100, 140, 90, 70, 150
OUT 7, 1
WAIT OUT 8, 1
```

DELAY
Retardo. Detiene la ejecución de un programa durante un tiempo especificado.

Sintaxis
- Sintaxis: DELAY [Retardo]

Explicación del comando
Este comando retarda la ejecución del programa un tiempo determinado. El tiempo de retardo para los controladores de la serie MR- C2000 es de 10ms y de un 1ms para los controladores de la serie MR-C3000.

Puede usar un valor, una constante, o una variable como valor del [retardo]

Ejemplo del comando
Controlador de la serie MR-C2000:
```
DELAY 10     'Retardo de 100ms. (10ms * 10 = 100ms = 0.1sec)
```

Controlador de la serie MR-C3000:
```
DELAY 500   'Retardo de 500ms. (1ms * 500 = 500ms = 0.5sec)
```

BREAK
Detiene la ejecución de un programa y pasa al modo de depuración. 2000

Sintaxis
- Sintaxis: BREAK

Explicación del comando
Detiene la ejecución del programa y entra en modo de depuración. Cuando se detiene la ejecución del programa, el estado de la memoria del controlador minirobot es enviada al PC. Compruebe que el PC y el controlador están conectados entre si. De otra manera, el programa se detendría y quedaría en ese estado. Se incluye mas información en "Explicación del programa roboBASIC".
El comando BREAK no funciona con los controladores de la serie MR-C3000. Si usa un controlador de la serie MR-C3000, puede seguir de manera sistemática la ejecución del código mediante la opción de depuración.

Explicación del comando
```
.............
BREAK        'Detiene la ejecución del programa
```

ACTION [no]
Reproduce los movimientos según la plantilla indicada en el valor (No.)

Sintaxis
- Sintaxis: ACTION [No.]

Explicación del comando
Reproduce los movimientos según la plantilla especificada. Hay un número máximo de 32 movimientos disponibles.

Ejemplo del comando
```
ACTION 3     'Reproduce los movimientos de la plantilla Nº.3.
ACTION 5     'Reproduce los movimientos de la plantilla Nr.5.
ACTION 23    'Reproduce los movimientos de la plantilla Nr..23
```

Nota: Este comando solo es para el controlador MR-C3024 y el Robot Robonova-I.

goto AUTO
Reproduce la plantilla incluida.

Sintaxis
- Sintaxis : goto AUTO

Explicación del comando
Comando para comenzar la ejecución del programa de la plantilla incluida.

Ejemplo del comando
```
Goto AUTO    'Pasa al programa de la plantilla .
```

Nota: Este comando es sólo para el controlador MR-C3024 y el Robot Robonova-I.

Capítulo 5

Explicación

de las señales digitales de entrada y

salida en roboBASIC

En el MR-C2000, existen 12 puertos digitales de E/S. En el MR-C3000, existen 40 puertos digitales de E/S. Estos puertos sirven para diferentes funciones. Consulte el "explicación de controladores" para obtener mayor información sobre los puertos de E/S.

IN()
Lee el valor de una señal digital desde un puerto.

Sintaxis
- Sintaxis: IN([número del puerto])

Explicación del comando
El valor de la señal que llega desde un puerto se almacena como variable. Los valores de entrada pueden ser 0 o 1. Se pueden usar variables de tipo Bit o Integer. En estos últimos solo se podrá disponer del último valor, 0 o 1. La manera más eficiente es usar variables de tipo Byte.

Ejemplo del comando
```
    DIM A AS BYTE
    A = IN(0)  'Lee una señal (interruptor o sensor) desde #0 y guarda el valor en la variable A.
```

OUT()
Envía una señal digital a un puerto.

Sintaxis
- Sintaxis: OUT [número del puerto], [valor a enviar]

Explicación del comando
Envía una señal desde el controlador a través de un puerto. Cuando se envía un valor de 0 (LOW), se emite una señal de 0V. Cuando se envía un valor de 1 (HIGH), la señal será de +5V. Pueden usarse números (0 o 1), constantes, y variables como [valor a enviar]. Sólo se pueden usar dígitos binarios como [valor a enviar] ya que solo estarán disponibles las señales 0 o 1 en el puerto de salida.

Ejemplo del comando
Este ejemplo sirve para comprobar los puertos de entrada/salida. Se ha conectado un pulsador al puerto #0 y un LED al puerto #3.

```
DIM A AS INTEGER

A = 0   'Inicializa la variable A
START: A = IN(0)    'Lee el estado del pulsador A.0 = IN(0) está disponible.

IF A = 1 THEN       'si el pulsador no se ha pulsado,
        OUT 3, 0    'apaga el LED
ELSE  'de otra manera,
        OUT 3, 1    'Enciende el LED
ENDIF
GOTO START          'check button again
```

BYTEIN()
Lee la señal de un puerto de entrada en formato de byte

Sintaxis
- Sintaxis: BYTEIN([número de puerto byte])

Explicación del comando
El puerto del controlador del robot puede leer/escribir tratando un conjunto de puertos como un byte al igual que el comando IN/OUT. En algunos casos, las señales de entrada/salida ha de ser tratadas de esta manera (aquí, usaremos el puerto #8 como puerto de byte)

Controlador de la serie MR-C2000:

Puerto de byte	Puerto
0	Puertos #0~#7 (El puerto #0 es el de menor peso en el byte)
1	Puertos #8~#11 (El puerto #8 es el de menor peso en el byte)

Controlador de la serie MR-C3000

Puerto de byte	Puerto
0	Puertos #0~#7 (El puerto #0 es el de menor peso en el byte)
1	Puertos #8~#15 (El puerto #8 es el de menor peso en el byte)
2	P. #16~#23 (El puerto #16 es el de menor peso en el byte)
3	P. #24~#31 (El puerto #24 es el de menor peso en el byte)
4	P. #32~#39 (El puerto #32 es el de menor peso en el byte)

Usando el comando BYTEIN, el valor de una señal leída a través del puerto (de byte) se almacena en una variable. La variable debe ser definida de tipo Byte o Integer.

Ejemplo del comando
```
A = BYTEIN(0) 'Se leen las señales de los puertos #0~ #7 y se guardan en la variable A
```

BYTEOUT
Envía una señal a un puerto en formato byte.

Sintaxis
- Sintaxis: BYTEOUT [Número del puerto byte], [valor de salida]

Explicación del comando
Envía señales a través del puerto tratado como byte. Puede usar valores, constantes o variables como [número del puerto]. Para el [valor de salida], debe usar números comprendidos entre 0~255, constantes o variables de tipo Byte.

Ejemplo del comando
```
BYTEOUT 0, 255      'Envía el valor 1 (5v) a los puertos #0~ #7
BYTEOUT 0, &h10101010
        "Envía el valor 1 (5v) a los puertos #1,#3,#5,#7
        ‘’Envía el valor 0 (0v) a los puertos #0,#2,#4,#6
```

INKEY
Lee la pulsación de una tecla (botón) a través del puerto de entrada. 2000

Sintaxis
- Sintaxis: INKEY ([número del puerto])

Explicación del comando
Cuando se pulsa un interruptor una sola vez, desde el punto de vista eléctrico y mecánico se pulsa cientos de veces. Este fenómeno recibe el nombre de “chattering”. El “chattering” puede producir errores de manera que se hace necesario añadir un circuito de protección. En el controlador del robot, el software dispone de esta función de protección por medio del software. Para manejar esta función debe usar el comando INKEY.

Ejemplo del comando
Salva el estado de un pulsador (pulsado o no= conectado al puerto # 0 en la variable A.

```
DIM A AS BYTE

A = INKEY(0)
```

STATE()
Lee el estado de un puerto de salida.

Sintaxis
- Sintaxis: STATE ([número del puerto])

Explicación del comando
Si fuese necesario averiguar el estado de un puerto de salida después de haber enviado datos, use la función STATE. No use la función IN.

Ejemplo del comando
Este es un programa de ejemplo para obtener el estado del puerto de salida #1.

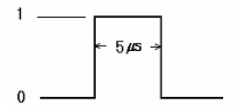
```
DIM A AS BYTE

OUT 1, 1
A = STATE(1)        ' A = 1
OUT 1,0
A= STATE(1)         ‘A = 0
```

PULSE
Envía una señal de pulso a un puerto de salida.

Sintaxis
- Sintaxis: PULSE [número del puerto]

Explicación del comando

Envía una señal de pulso a un puerto de salida durante 5 μs. Los pulsos se usan para enviar una señal a un dispositivo externo.



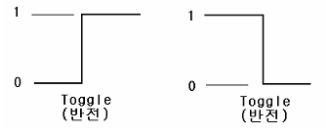
Puede usar números, constantes, o variables para indicar el [número del puerto]

Ejemplo del comando
PULSE 3 'Envía un pulso al puerto #3.

TOGGLE
Invierte el valor de la señal de un puerto de salida.

Sintaxis
- Sintaxis: TOGGLE [número de puerto]

Explicación del comando
Invierte la señal de salida de un puerto de salida. Si la señal del puerto es 0 (low), la señal se convertirá en 1 (HIGH).



Puede usar un número, una variable o una constante como valor del [número de puerto]

Ejemplo del comando
```
OUT 3, 1      'Envía señal "1" al puerto #3.
TOGGLE 3      'Invierte la señal del puerto #3.
```

KEYIN()
Lee varias teclas (teclado analógico). 3000

Sintaxis
- Sintaxis: KEYIN([número de puerto analógico], [número de teclas])

Explicación del comando
Este comando lee los valores de los 16 botones, recibidos a través de los puertos de conversión AD (puertos analógicos) de los controladores de la serie MR-C3000.
Se pueden usar números (0~7), constantes, y variables de tipo Byte para definir el [número del puerto analógico].
También puede usar números (1~16), constantes, y variables de tipo Byte para definir el [número de la tecla].
El rango de valores para un botón es de 0 a 16. 0 indica que la tecla no se ha pulsado. Los números del 1 al 16 indican que si se ha pulsado la tecla.

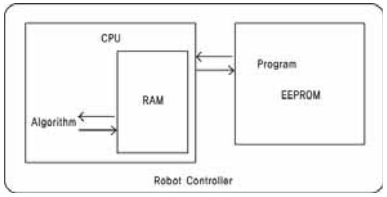
Ejemplo del comando
```
DIM K AS BYTE

K = KEYIN(0, 16)      'Lee los valores de las 16 teclas conectadas al puerto AD #0 y los almacena en K
```


Capítulo 6

Explicación de los comandos

relacionados con memoria

El controlador del robot tiene una CPU y una memoria, con lo que podríamos llamarlo un micro-ordenado. La ejecución en memoria juega un importante papel a la hora de guardar y ejecutar programas. La memoria externa para el controlador del robot, en forma de EEPROM, se usa para almacenar programas de usuario. La memoria para efectuar cálculos está en la CPU del robot.



La memoria interna se denomina como RAM, pero también es llamada registro, debido a las características especial de las CPUs de los controladores para robots. La memoria interna guarda relación con el número de variables que se pueden usar al crear un programa. Los controladores de la serie MR-C2000 disponen de 30 bytes para variables. Los de la serie MR-C3000 tienen hasta 256 bytes para este propósito. Las otras regiones de la memoria están reservadas para otros usos del controlador del robot.

La memoria externa guarda relación con el tamaño de los programas de usuario. La serie MR-C2000 está equipada con 4 Kbytes y la MR-C3000 tiene 12k, 32k, 64k dependiendo del modelo.

PEEK()
Lee el contenido de la memoria interna.

Sintaxis
- Sintaxis: PEEK ([Zona de la RAM])

Explicación del comando
La función PEEK recupera (lee) datos desde la memoria interna. Nunca use esta función, si no conoce perfectamente la estructura de la memoria.

Controladores de la serie MR-C2000:
Puede usar números 0~255, constantes o variables de tipo Byte.

Controladores de la serie MR-C3000:
Puede usar números 0~65535, constantes o variables de tipo Byte.

Ejemplo del comando
```
DIM A AS BYTE

A = PEEK(43)      'Vuelca el contenido de la zona 43 de la RAM en la variable A.
```

POKE
Escribe datos en la memoria interna.

Sintaxis
- Sintaxis: POKE [Zona de la RAM], [datos]

Explicación del comando
Puede usar el comando POKE para escribir datos en la memoria interna. En el MR-C2000, puede usar números del rango 0~255, constantes o variables del tipo Byte para definir la [Zona de la RAM] .
En el MR-C3000, puede usar números del rango 0~65535, constantes o variables para definir la [Zona de la RAM].

Puede usar números, constantes, o variables (de tipo Integer) para [datos].
Ejemplo del comando
POKE &h40, 100 'Escribe 100 en la zona (dirección) 40 de la RAM.

ROMPEEK()
Lee datos desde la EEPROM externa.

Sintaxis
- Sintaxis: ROMPEEK ([Zona de la ROM])

Explicación del comando
El controlador del robot usa una EEPROM para guardar programas u otros objetos. Las funciones ROMPEEK y ROMPOKE que controlan la memoria externas, pueden usarse para almacenar datos. Si la zona usada ya contuviese datos, se produciría un error general. Puede usar números, constantes o variables de tipo Byte para definir la [Zona de la ROM].

ROMPOKE
Escribe datos en la EEPROM externa.

Sintaxis
- Sintaxis: ROMPOKE [Zona de la ROM], [datos]

Explicación del comando
El controlador del robot usa una EEPROM para guardar programas u otros objetos. Las funciones ROMPEEK y ROMPOKE que controlan la memoria externa pueden usarse para guardar datos. Si la zona de la memoria ya contiene datos, puede producirse un error general. Puede usar números, constantes, o variables para definir la [Zona de la ROM]. Para definir los [datos] puede usar números entre 0 y 255, constantes o variables del tipo Byte.

Capítulo 7

Detalles del módulo LCD

en roboBASIC

El módulo LCD Diseñado para ser usado con el controlador del robot es el MR-16202. Conecte el módulo LCD al puerto #6 del controlador de la serie MR-C2000. El MR-C3000 tiene un puerto específico para el módulo LCD. Aquí se explican los comandos para manejar el módulo LCD y para mostrar cadenas de caracteres.

Módulo LCD MR-16202

LCDINIT
Inicializa el módulo LCD.

Sintaxis
- Sintaxis: LCDINIT

Explicación del comando
El módulo LCD ha de ser inicializado con el comando LCDINIT para evitar que aparezcan caracteres no deseados. Al inicializar el módulo LCD, se borrarán todos los caracteres y el cursor se posicionará en la esquina superior izquierda.

Ejemplo del comando
LCDINIT 'Inicializa el módulo LCD.

CLS
Borra caracteres del módulo LCD.

Sintaxis
- Sintaxis: CLS

Explicación del comando
Para borrar todos los caracteres mostrados en el módulo LCD, use el comando CLS. Al usar el comando CLS, se borrarán todos los caracteres y el cursor se posicionará en la esquina superior izquierda. Existen diferencias entre LCDINIT y CLS. El comando CLS solo borra caracteres, pero mediante el comando LCDINIT, se borra además todo el contenido interno, como variables, etc..

Ejemplo del comando
CLS 'Borra lo que se mostraba en el módulo LCD.

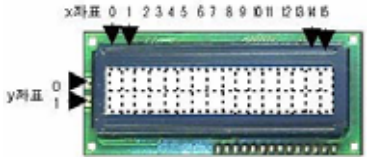
LOCATE
Apunta a la posición que ocupa un carácter en el módulo LCD.

Sintaxis
- Sintaxis: LOCATE [coordenada x], [coordenada y]

Explicación del comando
Defina las coordenadas x e y del módulo LCD con el comando LOCATE. Las coordenadas de un módulo LCD 16x2 se definen como se indica más abajo. Se pueden usar números, constantes, y variables para definir los valores de x e y, empezando por 0

Ejemplo del comando
LOCATE 0, 0 'posiciona el cursor en la esquina superior izquierda del módulo LCD.

LOCATE 4, 1 "posiciona el cursor en las coordenadas (4, 1) del módulo LCD.



PRINT

Envía caracteres al módulo LCD.

Sintaxis

- Sintaxis: PRINT “[cadena]”, [numero]/ “[cadena]]”, ...

Explicación del comando

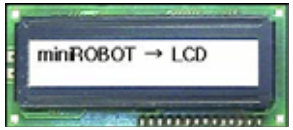
Para enviar un carácter a la posición actual del cursor, use el comando PRINT . La

[cadena] se delimita usando (“ “). El valor del [numero] debe estar entre 1 ~ 255 (0 no es válido). Se mostrarán en el visor los caracteres equivalentes al código ASCII.

Ejemplo del comando

CLS

PRINT "miniROBOT ", 126, "LCD"



Esto son algunos ejemplos del código ASCIII para un módulo LCD de 16x2 líneas. El código de los caracteres depende del módulo LCD que se esté usando.

FORMAT()
DEC()
HEX()
BIN()

Especifica el tipo de número a mostrar en el módulo LCD.

Sintaxis

- Sintaxis: FORMAT ([variable], [tipo de salida], [dígitos])

Explicación del comando

El módulo LCD sigue un formato determinados cuando muestra un carácter. El comando FORMAT debe seguir al comando PRINT.

[tipo de la variable]/[tipo de salida]	Dígitos	Expresión numérica
Tipo Byte /decimal	3	0 ~ 255
Tipo Byte /hexadecimal	2	00 ~ FF
Tipo Byte/binario	8	00000000 ~ 11111111
Tipo Integer /decimal	5	0 ~ 65535
Tipo Integer /hexadecimal	4	0000 ~ FFFF
Tipo Integer /binario	16	0000000000000000 ~ 1111111111111111

Ejemplo del comando

DIM A AS BYTE
DIM B AS INTEGER

LCDINIT
A = 100
B = 20000
LOCATE 0, 0PRINT FORMAT(A, DEC, 4) LOCATE 0, 1
PRINT FORMAT(B, HEX)

CSON()
CSOFF()
Muestra/oculta el cursor en el módulo LCD.

Sintaxis
- Sintaxis: CSON / CSOFF

Explicación del comando
El comando CSON/CSOFF mostrará u ocultará el cursor en el módulo LCD. Por regla general, al inicializar el módulo el cursor no se muestra.

Ejemplo del comando
```
LCDINIT
CSON
PRINT "CURSOR ON"
```

CONT
Ajusta el contraste del módulo LCD.

Sintaxis
- Sintaxis: CONT [valor de contraste]

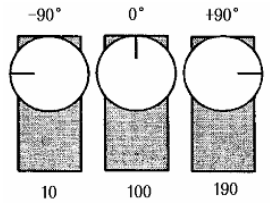
Explicación del comando
El módulo LCD está retroiluminado. Los caracteres se muestran con color negro. Mediante el comando CONT, se puede ajustar la intensidad del color. Puede usar número, constantes, y variables para definir el [valor del contraste]. A medida que el [valor del contraste] aumenta, los caracteres irán "engordando". El valor inicial es 7.

Ejemplo del comando
```
LCDINIT

CONT 10
PRINT "miniROBOT"
```

Capítulo 8

Explicación de los comandos para el control del motor en roboBASIC

El controlador del robot puede manejar servos y motores DC. En el caso de los motores DC, el controlador puede manejar la velocidad, dirección, y la para del motor usando ordenes de entrada y salida digitales.
El rango de movimientos del servo van de -90° a +90°. Para manejar los servos en roboBASIC, los grados se expresan mediante valores entre 10 y 190, ya que el controlador no gestiona número negativos.



Varios servos:

HS-311 HS-5645MG HSR-8498HB AIMOTOR

ZERO
Define el punto cero del servo.

Sintaxis
Controlador de la serie MR-C2000:
- Sintaxis: ZERO [punto 9 del motor], [punto 1 del motor], ..., [punto 5 del motor]

Controlador de la serie MR-C3000:
- Sintaxis : ZERO [puntero de grupo], [punto n del motor], ...

Explicación del comando
El punto cero de cada servo depende del propio servo. Esto es debido a las desviaciones de cada producto. Algunos puntos cero pueden ser 99 o 98, otros puntos cero pueden ser 101 o 102. Este tipo de error puede ser ajustado mediante el comando ZERO. Una vez configurado el punto cero para cada servo, se tomará como punto de referencia para el comando MOVE.
El punto cero configurado se almacenará en la EEPROM para evitar que se pierda al desconectar la alimentación.

Definición del punto cero en la serie MR-C2000:
1) Defina el punto cero de todos los servos a 100, para borrar datos antiguos.
2) Defina la dirección del motor como normal.
3) Apague y vuelva a encender.
4) Mueva el motor al punto cero (centro) usando funciones Online
5) Guarde esta posición como punto cero, usando el comando Zero

EX1: elimina el valor antiguo del punto cero

```
ZERO 100,100,100,100,100,100
DIR   1,1,1,1,1,1                    'pone la dirección del motor en normal
```

EX2: Configura de nuevo el punto cero. Guarda la posición (centro)

```
DIR   1,1,1,1,1,1
ZERO 100, 101, 99                    ' Establece el punto cero del motor de 3 servos
ZERO 102, 100, 100, 99, 101, 100     ' Establece el punto cero del motor de 6 servos
```

En los controladores de la serie MR-C2000, el valor del punto cero debe estar comprendido entre 90~110.

Definición del punto cero en la serie MR-C3000:
Cuando se define el punto cero en un controlador MR-C3000, se debe declarar un grupo.

```
ZERO G8B, 80, 120, 115, 80, 117, 88, 95, 120      'Define el punto cero del grupo 8B(servos
                                                    #8~#15)
```

En los controladores de la serie MR-C2000, el valor del punto cero debe estar entre 80~120.

MOTOR
Define el servo a utilizar.

Sintaxis
Controlador de la serie MR-C2000:
- Sintaxis: MOTOR [número de motor]

Controlador de la serie MR-C3000:
- Sintaxis: MOTOR [número de motor] / [Grupo]

Explicación del comando
Selección del servo en la serie MR-C2000:
En los controladores de la serie MR-C2000, existen 6 puertos para los servos (#0~#5). Los número que puede usar para definir un motor son 0~5. Si quiere usar todos los servos a la vez, use el número 6.

Si no se define un número para el servo, el motor no funcionará. Solo puede usar número del 0~6 como valor del [número de motor].

```
Ex1) MOTOR 6      'Se usan todos los motores (#0~#5)
Ex2) MOTOR 2      'Se usa el motor #2
```

Selección de un servo en la serie MR-C3000:

En los controladores de la serie MR-C3000, existen 32 puertos para 32 servos. Cada motor puede ser definido mediante [número de motor]. Los grupos de motores se definen con [grupo de motores]. Puede usar números, constantes o variables de tipo Byte para [número de motor].

Ejemplo del comando

```
Ex 1)  MOTOR 0              'Se usará el servo #0
Ex 2)  MOTOR G6A            ' Se usará el grupo de servos 6A (#0~#5).
       MOTOR G6C            ' Se usará el grupo de servos 6C (#12~#17)
Ex 3)  MOTOR G8A            ' Se usará el grupo de servos 8A (#0~#7)
Ex 4)  Seleccionando el servo mediante una variable
       DIM I AS BYTE
       FOR I = 0 TO 31      'Se usaran los servos (#0~#31) mediante la variable I'
       MOTOR I
       NEXT I
Ex 5)  MOTOR G24            'se usará el grupo de servos 24 (#0~#23)
Ex 6)  MOTOR ALLON          'Se usarán todos los servos.
```

MOTOROFF
Desconecta un servo.

Sintaxis
Controlador de la serie MR-C2000:
- Sintaxis: MOTOROFF [número del motor]

Controlador de la serie MR-C3000:
- Sintaxis: MOTOROFF [número de motor] / [grupo]

Explicación del comando
El comando MOTOROFF es el mismo que el comando MOTOR.

MOVE
maneja varios servos a la vez.

Sintaxis
Controlador de la serie MR-C2000:
-Sintaxis: MOVE [ángulo del motor0], [ángulo del motor1], ..., [ángulo del motor5]

Controlador de la serie MR-C3000:
-Sintaxis: MOVE [grupo], [ángulo del motor n], ...

Explicación del comando
Comando Move en el controlador de la serie MR-C2000:
El comando MOVE hace que un servo se desplace hasta el ángulo deseado. Durante su uso, la función PWM se deshabilita. El rango del [ángulo del motor] debe ser entre 10 ~190. Si quiere mover los servos #1,#3, y #4, la sentencia debería ser como esta:
MOVE 60, , 100, 120
Si quiere mover solo el motor #2, se parecería a este ejemplo.
MOVE, 140
Este proceso pude resultar complicado, especialmente si se quieren mover 6 servos a la vez.
. Sería más sencillo si se usara un "control de servos en tiempo real".
Si se usa un motor DC, 100 significa 'stop', 190 velocidad máxima con rotación inversa y 10 velocidad máxima con rotación normal.

Use el comando WAIT, para usar el motor después de enviar este comando.

Ejemplo del comando

```
MOVE 100, 50, 140, 120, 80, 40
MOVE 120, , , 160
MOVE , 70, 100
MOVE , , , , , 100
```

Comando Move con un controlador de la serie MR-C3000:
En el controlador de la serie MR-C3000, los puertos para los servos y para PWM sin diferentes. De manera que puede usar el comando Move y PWM al mismo tiempo.

Ejemplo del comando

```
EX 1)
    MOVE    G6A, 85, 113, 72, 117, 115, 100
    MOVE    G6C, 75,    , 96, 123,      , 122
    MOVE    G8A, 85, 113, 72, 117, 115, 100, 95, 45

EX 2)
    MOVE    G24, 85, 113, 72, 117, 115, 100
Es lo mismo que;
    MOVE24  85, 113, 72, 117, 115, 100
```

SPEED
Configura la velocidad de un servo

Sintaxis
- Sintaxis: SPEED [velocidad del motor]

Explicación del comando
El comando SPEED configura la velocidad de un servo usado por el comando MOVE. En los controladores de la serie MR-C2000 , puede usar números, o constantes entre 1~15 como valor de [velocidad del motor]. En los

controladores de la serie MR-C3000, puede usar variables de tipo Byte . El valor por defecto es 3. Poner una velocidad demasiado alta, es peligroso tanto para el robot como para el usuario.

Ejemplo del comando

Ex 1)

```
SPEED 7                     'Define la velocidad del motor a 7.
```

Ex 2)

```
DIM STEP_SPEED AS BYTE      'Declara STEP_SPEED (Variable).
STEP_SPEED = 15             'Asigna 15 a STEP_SPEED (Variable)
SPEED STEP_SPEED            'Define SPEED como STEP_SPEED
```

ACCEL

Configura la aceleración de un servo

Sintaxis

- Sintaxis: ACCEL [aceleración del motor]

Explicación del comando

El comando ACCEL configura el rango de aceleración desde 0 a la velocidad configurada.
[Aceleración del motor] usa números y constantes entre 0 y 15. El valor normal es 3.Un número mayor aumenta la aceleración del servo.
Cuando maneje el servo por primera vez, éste se posicionará rápidamente. Para disminuir esta velocidad inicial, es mejor usar los comandos ACCEL y SPEED.
En los controladores de la serie MR-C3000, no se puede usar el comando ACCEL.

Ejemplo del comando

```
ACCEL 7                     'Configura la aceleración del servo a 7.
```

DIR

Define el sentido de rotación de un servo.

Sintaxis

Controlador de la serie MR-C2000:

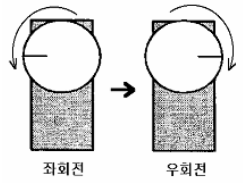
- Sintaxis: DIR [dirección motor 0], [dirección motor 1], ..., [dirección motor 5]

Controlador de la serie MR-C3000:

- Sintaxis: DIR [grupo], [dirección motor n], ...

Explicación del comando

Un servo girará a la izquierda cuando el ángulo sea inferior a 100 (ángulo standard) y girará a la derecha cuando el ángulo sea superior a 100 (ángulo standard). En la imagen de abajo, el servo girará a la izquierda 10 grados (sentido normal).



[dirección del motor] usa constantes o número como 0 (sentido inverso/a la izquierda) o 1 (sentido normal/a la derecha). El valor por defecto es 0. por ejemplo, cuando se usan 4 servos si se omite el servo #, como “DIR 0, 1, , 0”), girará en el sentido normal.

Ejemplo del comando

Ejemplo para un controlador de la serie MR-C2000:

```
DIR 0, 1, 1, 0, 1, 0
DIR , , 0
```

Ejemplo par aun controlador de la serie MR-C3000

```
DIR G8A, 0, 1, 0, 0, 1, 0, 0, 0
DIR G8B, 1, 0, 1, 1, 0, 1, 1, 1
```

PTP — Point to Point

Configura la opción On/OFF para el control simultaneo de los servos

Sintaxis

Controlador de la serie MR-C2000:

- Sintaxis: PTP [valor a configurar]

Controlador de la serie MR-C3000:

- Sintaxis: PTP [SETON/SETOFF/ALLON/ALLOFF]

Explicación del comando

En el caso de múltiples movimientos o movimientos con diferentes ángulos, los puntos finales de los servos serán distintos entre sí. Así, en el caso de un brazo robótico o cualquier otro robot operado por servos, los movimientos serían inestables.
En la ingeniería de robots, existe una teoría llamada Point-to-Point, que puede calcular los puntos finales de los servos (tiempos) y termina simultáneamente todos los movimientos logrando un control más preciso.

Los controladores de la serie MR-C usan este método de control Point-to-Point mediante el uso del comando PTP.

Control PTP en el MR-C2000

El [valor configurado] es 0 (cancel) o 1(set up) (numero o constante) cuando se usan dos servos (No.1 y No. 2). Vea los siguientes ejemplos:

- Ejemplo de un movimiento normal

```
PTP 0
MOVE 100, 100
MOVE 110, 120
```

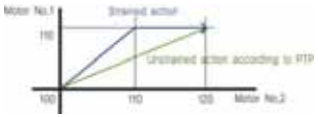
Descripción: El motor N. 1 se mueve 10 grados y el motor N. 2 lo hace 20 grados. Los dos se mueven a la misma velocidad simultáneamente 10 grados, y después, el N. 2 se moverá los 10 grados restantes.

- Movimiento suave usando la función PTP

```
PTP 1
MOVE 100, 100
MOVE 110, 120
```

Descripción: El motor N. 1 se mueve 10 grados y el motor N. 2 motor lo hace 20 grados pero el N. 1 se mueve a la mitad de velocidad del N.2. Ambos servos arrancan y se detienen a la vez.

Vea e gráfico inferior para ver las diferencias entre usar el comando “PTP” (flecha verde) o no usarlo (flecha azul).



Fíjese como con el movimiento normal, los motores N.1 y N.2 se mueven hasta llegar al ángulo 110 simultáneamente, y después, solo el N.2 se posiciona a 120. Sin embargo, al usar el comando “PTP", al calcular la diferencia del ángulo de 10º para N.1 y 20º para N.2, se consigue una finalización de movimientos acompasada.

Control PTP con el controlador de la serie MR-C3000)
Puede usar múltiples servos con el controlador de la serie MR-C3000s. Puede ajustar el funcionamiento PTP para controlar todos los servos o para grupos individuales.

- PTP SETON (PTP setup): Configura PTP para usar grupos.
- PTP SETOFF (PTP cancel): Cancela el uso de grupos por PTP.
- PTP ALLON (PTP all setup): Configura PTP para usar todos los servos.
- PTP ALLOFF (PTP all cancel): Cancela el uso de todos los servos por PTP.

En los controladores de la serie MR-C3000 , al usar el comando “WAIT” al final del movimiento de cada grupo, todos las servos de ese grupo acaban de moverse al mismo tiempo.

SERVO
Se maneja un servo

Sintaxis
- Sintaxis: SERVO [N. motor],[ángulo]

Explicación del comando
Este comando define el ángulo deseado para el motor. En los controladores de la serie MR-C2000, se cancela la función PWM. El [N. motor] es el número del puerto del controlador. El [ángulo] está comprendido entre 10 y 190, debiendo usar un número, una constante o una variable de tipo byte.

Ejemplo del comando
Ex 1)
```
SERVO 1,130          ' opera el motor No.1 y lo mueve a 130º
```
Ex 2)
```
DIM I AS BYTE

FOR I = 10 TO 190
    SERVO 4, I DELAY 100
    NEXT I
```

PWM
Pulse Width Control (Control de amplitud de impulsos)

Sintaxis
- Sintaxis: PWM [N. motor],[valor de la amplitud de impulsos]

Explicación del comando
Control de PWM con el MR-C2000:
Con los controladores de la serie MR-C2000, se usan a la vez, los puertos de servos del controlador y los puertos PWM . Por tanto, [N. motor] oscila entre 0 y 5.
Si se envía el comando PWM con un valor para la amplitud, la función servo se cancela.
(Aviso: En un programa, los comandos SERVO o MOVE no pueden usarse con el comando PWM.)

Rango	Valor de amplitud	Rango	Valor de amplitud
0	0	***60***	153
10	25	***70***	178
20	51	***80***	204
30	77	***90***	229
40	102	***100***	254
50	127		

```
PWM 3, 127          ' PWM emite el 50% del rango al puerto del motor No.3.
```

Control de PWM con un controlador de la serie MR-C3000s
El control del servo y del PWM no usan los mismos puertos. Hay 3 puertos PWM (No.0~No.3) instalados en los controladores de la serie MR-C3000s (454Hz es la frecuencia PWM en la serie MR-C3000).

```
PWM 0, 120          ‘salida de 120 en el puerto N.0 de PWM.
```

FASTSERVO
Mueve un servo a mayor velocidad 2000

Sintaxis
- Sintaxis: FASTSERVO [N. motor],ángulo del motor]

Explicación del comando
Este comando posiciona un servo en el ángulo especificado a la mayor velocidad posible. Solo los controladores de la serie MR-C2000 usan este comando. El [N.motor] es el puerto del servo y [ángulo del motor] es el ángulo deseado. Puede usar números y constantes entre 10 y 190 para definir el [ángulo del motor].

Ejemplo del comando
```
FASTSERVO 2,190   'Posiciona el motor No.2 a un ángulo de 190º lo más rápido posible.
```

HIGHSPEED
Pone un servo a alta velocidad 3000

Sintaxis
- Sintaxis: HIGHSPEED [SETON/SETOFF]

Explicación del comando
Este comando activa/desactiva la alta velocidad de un servo en los controladores de la serie MR-C3000s. Alta velocidad es 3 veces la velocidad normal.

- HIGHSPEED SETON: Activa el modo alta velocidad en controladores de la serie MR-C3000s.
- HIGHSPEED SETOFF: Desactiva el modo alta velocidad en controladores de la serie MR-C3000s. (Vuelve al modo de velocidad normal).

Ejemplo del comando
```
HIGHSPEED SETON          Configura al alta velocidad.
```

MOVEPOS — Move position
POS — Motor position
Define un punto de movimiento o la posición de un servo 3000

Sintaxis
- Sintaxis: MOVEPOS [etiqueta de línea]

……
[etiqueta de línea]: POS [grupo], [ángulo motor n], …

Explicación del comando
Mientras la posición del robot) viene dada por otros comandos (el comando 'move', en los controladores de la serie MR-C3000, se maneja con los comandos 'POS' (motor position) y [etiqueta de línea] con el comando 'MOVEPOS'. Con los comandos 'MOVEPOS' y 'POS', podrá modificar fácilmente el programa roboBasic.

Ejemplo del comando

```
……
MOVEPOS   POS01 'Comando move 'POS' parte de la etiqueta de posición 'POS01'
……
POS01:    POS G6A, 10, 32, 15, 120, 78, 93
POS02:    POS G6A, 67, 47, 32, 153, 23, 33
POS03:    POS G6A, 34, 37, 122, 162, 84, 28
```

FPWM — Frecuencia de pulso

Emite una señal PWM (La frecuencia puede ser variable) 3000

Sintaxis
- Sintaxis: FPWM [puerto], [frecuencia], [rango]

Explicación del comando
Se modifica la frecuencia del PWM y los impulsos emitidos por el controlador de la serie MR-C3000s.
Puerto: De 0 a 2
Frecuencia: Desde 1(baja frecuencia) ~5 (alta frecuencia) Rango: DE 0 a 255

Ejemplo del comando

```
FPWM 0, 1, 127        'Envía una señal PWM (al 50% rango (127) y de baja frecuencia en el puerto N.0 de PWM
                      del controlador de la serie MR-C3000)
```

MOVE24

Mueve el motor de los 24 servos 3000

Sintaxis
- Sintaxis: MOVE24 [Ángulo del motor 0],…….,[Ángulo del motor]

Explicación del comando
Puede manejar grupos de servos usando el comando "MOVE" en los controladores de la serie MR-C3000s. Los comandos "MOVE24" y "MOVE G24" se usan para manejar los 24 servos de manera simultánea. Es un buen comando para manejar un robot con 16 hasta 24 servos.

Ejemplo del comando

```
MOVE24    100, 45, 67, 44, 132, 122, , , , 76, 81, 90
```

INIT — Initial

Define la posición inicial del robot 3000

Sintaxis
- Sintaxis: INIT [grupo designado], [ángulo del motor n],……

Explicación del comando
Si el controlador de la serie MR-C3000 está instalado en un robot, todos los servos se colocan en una posición inicial de "100" durante el encendido. Podrían provocarse daños en el robot.

Para evitar posibles daños, puede definir la posición inicial con un valor distinto a "100".
Para servos analógicos (serie HS), use el comando "INIT".
Para servos digitales de robot (serie HSR-series), use el comando "GETMOTORSET".

Ejemplo del comando

```
INIT G8A, 100, 45, 67, 44, 132, 122, 76, 81
```

MOTORIN() — Motor input()

Lee el ángulo actual de un servo. 3000

Sintaxis
- Sintaxis: MOTORIN ([N. motor])

Explicación del comando
Con este comando, puede leer el valor de la posición actual de cualquier servo de robot (serie HSR) conectado al controlador de la serie MR-C3000 y controlarlo. Puede usar un número entre 0 y 31 como valor para [N. motor].
Conectado al controlador: Se obtienen valores entre 10 y 190. Sin conectar: ángulo del motor = 0

Ejemplo del comando

```
DIM   S0   AS   BYTE

MOTOR 0                'Usa el servo No. 0.
S0 = MOTORIN(0)        'Guarda el valor del servo No.0 en la variable S0.
```

AIMOTOR — AI motor set up

Usa el motor AI 3000

Sintaxis
- Sintaxis: AIMOTOR SETON/SETOFF/INIT/[N. motor] / [grupo]

Explicación del comando
El motor AI está fabricado por Megarobotics. El motor AI tiene instalado un micro-chip de control, que puede comunicarse con los controladores de la serie MR-C vía RS232. El motor AI se controla por los controladores de la serie MR-C de la misma manera que un servo normal.
- Control del ángulo del motor
- Estado actual del motor y control del par mediante control PDI (PGAIN, DGAIN)

Los motores AI se pueden conectara los puertos No. 0 a No. 30 (31 puertos en total),
[N. motor] : apunta a uno de los motores,
[grupo] : apunta a un grupo de motores

El procesado de las ordenes es parecido al del comando "MOTOR". Puede usar un número, una constante, o una variable de tipo byte como valor de [N. motor].
Cuando use un motor AI, la declaración "use AI motor" es imprescindible, pero no es necesaria para los otros servos.
- AIMOTOR SETON: Define el uso de un motor AI.
- AIMOTOR SETOFF: Se cancela el uso de un motor AI.
- AIMOTOR INIT: Mueve lentamente el motor AI a su posición inicial.

Ejemplo del comando

```
AIMOTOR INIT          'Inicializa el motor AI
AIMOTOR SETON         'Define el uso del motor AI
AIMOTOR 0             'Usa el motor AI No. 0
AIMOTOR G6B           'Usa el grupo de motores 6B (N.6-11)
```

AIMOTOROFF — Deja de usar el motor AI

Detiene el uso del motor AI 3000

Sintaxis
- Sintaxis: AIMOTOROFF [N. motor] / [grupo]

Explicación del comando
Cancela el comando "AIMOTOR".
Funciona igual que el comando "MOTOROFF".

Ejemplo del comando
```
AIMOTOROFF 0          'Cancela el motor AI No.0.
AIMOTOROFF G6B        'Cancela el grupo de motores 6B(No.6-11).
AIMOTOR SETOFF        'Deja de usar el motor AI.
```

AIMOTORIN() Lee el valor del motor AI ()

Lee el valor actual de un motor AI 3000

Sintaxis
- Sintaxis: AIMOTORIN ([N. motor])

Explicación del comando
Con este comando se puede leer el valor actual de un motor AI y controlarlo, siempre que este conectado aun controlador de la serie MR-C3000s.
Puede usar número de 0 a 30, como constantes para el valor de [N. motor].
Si está conectado: Podrá obtener valores entre 10 y 190.

Sin conexión: Ángulo del motor = 0

Ejemplo del comando
```
DIM   AI5   AS   BYTE

AIMOTOR INIT
AIMOTOR SETON
AIMOTOR 5             'Se usa el motor AI No.5
AI5 = AIMOTORIN(5)    'Guarda el valor del motor AI No.5 AI en la variable AI5
```

GETMOTORSET Lee y pone una entrada

Lee el estado de un servo y lo mueve a un punto. 3000

Sintaxis
- Sintaxis: GETMOTORSET [grupo], [motor n a usar],…

Explicación del comando
En un sistema de comunicación interactiva, se puede leer la posición actual de un servo digital (serie HSR) conectado a un controlador de la serie MR-C3000.
Si el controlador de la serie MR-C3000 está instalado en un robot, todos los servos retornan a un punto inicial "100" durante el encendido y podrían causar desperfectos en el robot.
Para evitar estos daños, se puede establecer una posición inicial distinta de "100". Lea la posición inicial antes de alimentarlos y ejecute la acción deseada usando el comando "move".

Para [motor n a usar], use "0" o "1". Si escoge "1", se leerá el valor actual del servo y se dejará el robot donde estaba. Si utiliza "0", se tomarán los valores por defecto del controlador que es 100.

Ejemplo del comando
```
GETMOTORSET G8A, 1, 1, 1, 1, 0, 0, 0, 0
      'No. 0, 1, 2, 3 los motores mantienen su posición actual al recibir alimentación.
      'No. 4, 5, 6, 7 los motores se mueven al punto100 al recibir alimentación.
```


Capítulo 9

roboBASIC

Comandos para el control de música

Los controladores de la serie MR-C tienen la capacidad de reproducir música y tonos de aviso. Para realizar estas tareas, se necesita un piezo separado. El cableado del piezo consiste en un cable rojo (+) y una cable blanco para las señales.
Si desea un sonido más alto y nítido, podría conectar un altavoz, pero necesitará un circuito de potencia o un amplificador.

Conexión al controlador de la serie MR-C2000:
El piezo se conecta al puerto No.8 del controlador de la serie MR-C2000. El cable (+) del piezo se conecta a VCC del puerto No. 8 y el cable (-) se conecta a SIG (signal) del puerto No.8.

Conexión al controlador de la serie MR-C3000:
El piezo se conecta al puerto No.28 del controlador de la serie MR-C3000 . El cable (+) del piezo se conecta a VCC del puerto No. 28 y el cable (-) se conecta a SIG (signal) del puerto No.28.

Nota: El controlador MR-C3024 incluye un piezo-eléctrico.

BEEP

Emite un pitido 2000

Sintaxis
- Sintaxis: BEEP

Explicación del comando
Para emitir un pitido (tono de aviso) usando un controlador de la serie MR-C2000 , use el comando "BEEP". El uso de un zumbador para emitir avisos solo es posible si lo conecta a los puertos de salidas normales al usar el comando "OUT". *El controlador de la serie MR-C3000 usa el zumbador mediante el comando "OUT".*

Ejemplo del comando
```
BEEP  'Emite un tono de aviso
```

SOUND

Emite un sonido 2000

Sintaxis
- Sintaxis: SOUND [tono, [duración], [tono], [duración], ...

Explicación del comando
En el controlador de la serie MR-C2000s, se puede definir la frecuencia de la señal y el retardo del puerto donde está conectado el piezo.
Valores: Desde 1 a 254
Fíjese en la tabla inferior

Valor	Frecuencia (Hz)	Valor	Frecuencia (Hz)	Valor	Frecuencia (Hz)
1	38,86k	***70***	800	***160***	389
2	23,81k	***80***	775	***170***	365
5	11,11k	***90***	689	***180***	344
10	5,88k	***100***	621	***190***	327
20	3,00k	***110***	565	***200***	311

30	2,00k	***120***	518	***210***	295
40	1,54k	***130***	478	***220***	283
50	1,23k	***140***	444	***230***	270
60	1,00k	***150***	413	***240***	260

Tiempo	Duración (11msec)
0,5s	45
1s	90
2s	180

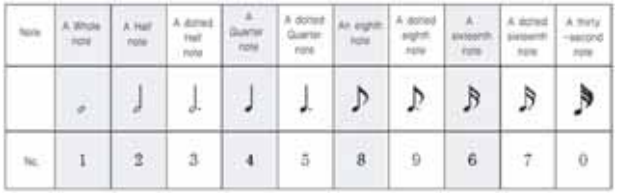
Ejemplo del comando
SOUND 60, 90, 60, 180, 30, 45 'Genera una frecuencia de 1KHz durante 1seg, 2seg y una de 2KHZ durante 0.5sec.

PLAY
Reproduce música 2000

Sintaxis
- Sintaxis : PLAY "[cadena musical]"

Explicación del comando
RoboBASIC y los controladores de la serie MR-C pueden reproducir música. Para hacerlo, debe añadir las "notas" a la [cadena musical]. Vea la tabla inferior:

Cadena musical	Describe
C	„Do“
D	„Re“
E	„Mi“
F	„Fa“
G	„Sol“
A	„La“
B	„Si“
T	Control de tempo, ritmo y velocidad
L	Octava baja
M	Octava media
H	Octava alta
#, +	Sube un tono (#)
$, -	Baja un tono (♭)
P, ,(Rest von a)	Pausa
<	Baja una octava
>	Sube una octava

T significa tempo y el tempo base tiene valor 7.
Se puede ajustar de 1 a 0 (0 significa 10). 1 es el tempo más rápido y 0 el más lento.
Hay 3 niveles de octavas con un controlador de la serie MR-C2000, se llaman baja, media y alta.
Para ajustar la duración de un tono, use números de 0 a 9 y consulte la tabla inferior:

Si no se especifica una duración, se tomará la última especificada.
"4CDEF8G" significa reproducir Do, Re, Mi y Fa con una corchea de Sol como octava nota.
El valor por defecto del comando "PLAY" es la media octava. La corchea, el Do, y tempo 7 tendrán los mismos valores para tempo y octava que el último valor utilizado.

① La duración del tono se especifica [4Do 8Mi 6Fa].
② El símbolo # u otras notaciones se anteponen a la nota [#Do $Mi +Fa - Sol 4#Do #4Do].

Use elcomando "MUSIC" en lugar de "PLAY" si usa un controlador de la serie MR-C3000s

Ejemplo del comando
PLAY "M4GGAA GGE GGEED" PLAY "M4GGAA GGE GEDEC"

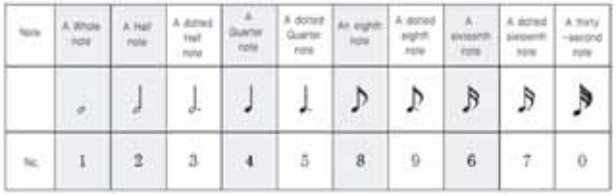
MUSIC
Reproduce música con un piezoeléctrico. 3000

Sintaxis
- Sintaxis: MUSIC "[cadena musical]"

Explicación del comando
El lenguaje roboBASIC y los controladores de la serie MR-C disponen de una función para reproducir música. Para reproducirla, debe añadir los datos a [cadena musical]. Consulte la tabla inferior:

Cadena musical	Describe
C	„Do“
D	„Re“
E	„Mi“
F	„Fa“
G	„Sol“
A	„La“
B	„Si“
[	Acorta el tono 1.5 veces para un grupo de notas
]	Alarga el tono1.5 veces para un grupo de notas
O	Selecciona una octava
M	Selecciona 3 octavas
.	Alarga el tono 1.5 veces
#, +	Sube un tono (#)
$, -	Baja un tono (♭)
P, ,(Rest von a)	Pausa
<, L	Baja una octava
>, H	Sube una octava

El controlador de la serie MR-C3000 tiene siete octavas. Para ajustar la duración de un tono, use los números del 0 al 9. Consulte la tabla inferior:

En caso de notas con puntuación, escríbalas con un número o "." en la [cadena musical].

Tiene que usar el comando "PLAY" en vez de "MUSIC" si usa un controlador de la serie MR-C2000 .

El comando "TEMPO" está disponible por separado en el controlador de la serie MR-C3000s.
Ejemplo del comando
MUSIC "O34GGAA GGE GGEED"
MUSIC "O3GGA4.A GGE GEDEC"

TEMPO

Configura el tempo de la música 3000

Sintaxis

- Sintaxis: TEMPO [valor]

Explicación del comando

Configura el tempo de la música a reproducir con el comando "MUSIC" al usar un controlador de la serie MR-C3000s.

Capítulo 10

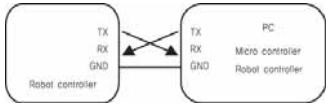
roboBASIC

Comandos para comunicaciones con el exterior

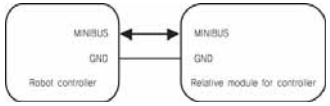
Comunicaciones externas en el controlador de la serie MR-C2000):

Puede usar dos tipos de comunicaciones con el exterior al trabajar con un controlador de la serie MR-C2000s. Una es la comunicación serie RS232 y la otra es miniBUS.

En comunicaciones en serie, puede comunicarse de manera interactiva con un ordenador personal (compatible RS232) o con otro controlador MR-C. Además, también puede comunicarse mediante cable o por wireless. Necesitará tres cables para comunicarse por RS232 (conectando transmisión (TX), recepción (RX) y tierra (GND)).



Si lo conecta a un ordenador, necesitará un convertidor de voltaje (MAX232). Si usa miniBUS, las señales BUS y tierra (GND) se usan para comunicaciones interactivas.



MiniBUS solo usa un cable para comunicaciones interactivas. Debe respetar algunas reglas especificas. El módulo LCD es un ejemplo de transmisión de datos usando minibus. El controlador instalado en el LCD solo recibe información.

Comunicaciones externas usando un controlador de la serie MR-C3000:

Puede comunicarse a alta velocidad (RS232) con equipos externos usando los controladores de la serie MR-C3000. Sin embargo, las comunicaciones que usan miniBUS no están soportadas en la serie MR- C3000. Ambos tipos de controladores tienen una velocidad máx. RS232 de 115,200bps.

RX

El puerto RX recibe una señal RS232. 2000

Sintaxis

RX [Velocidad del puerto], [Variable de recepción], [Etiqueta rutina recepción]

Explicación del comando

Se pueden recibir datos vía RS232, usando el puerto No. 9 de los controladores de la serie MR-C2000.
La [velocidad del puerto] se representa mediante los números 1 a 4 donde cada uno de ellos tiene un valor determinado, que se explica en la tabla inferior.

Número	Configuración del puerto
1	1200bps, 8Bit datos, sin paridad, 1 bit Stop
2	2400bps, 8Bit datos, sin paridad, 1 bit Stop
3	2400bps, 8Bit datos, sin paridad, 1 bit Stop
4	4800bps, 8Bit datos, sin paridad, 1 bit Stop

[Variable de recepción] es la variable donde se guardarán los datos recibidos. Solo se permiten variables de tipo byte.
[Etiqueta rutina de errores] nombre de la rutina de tratamiento de errores. Por ejemplo
Si se usan las comunicaciones y el buffer está vacío. Todos los programas que esperen recibir datos vía RS232 pueden usar la siguiente sintaxis.

Retry:
RX 4, A, Retry
Para recibir señales RS232 con un controlador de la serie MR-C3000s, debe usar ERX (Datos).

Ejemplo del comando

En este ejemplo, se recibe vía RS232 el carácter ASCII, &h80 (16 en analógico), desde un terminal externo, el LED del puerto cero se enciende, todos los demás apagados.

```
DIM   A   AS   BYTE

Retry:   RX 4, A, Retry
IF A = &h80 THEN OUT 0, 0
ELSE
OUT 0, 1
ENDIF
GOTO Retry
```

TX

El puerto TX emite una señal RS232 2000

Sintaxis

TX [Velocidad del puerto], [Datos]

Explicación del comando

Se pueden enviar datos vía RS232, usando el puerto No. 10 de un controlador de la serie MR-C2000.
La [velocidad de puerto] se representa mediante número del 1 a 4 donde cada uno, tiene un valor determinado, y configura las propiedades del puerto.

Número	Config. puerto
1	1200bps, 8Bit datos, Sin paridad, 1 bit Stop
2	2400bps, 8Bit datos, Sin paridad, 1 bit Stop
3	2400bps, 8Bit datos, Sin paridad, 1 bit Stop
4	4800bps, 8Bit datos, Sin paridad, 1 bit Stop

[Datos] es el valor de los datos transmitidos por el puerto TX. Puede usar números, constantes y variables. Para enviar la letra "A", debería enviar el código ASCII correspondiente a la letra "A. Fíjese en el ejemplo inferior.

```
DIM I AS BYTE
I = "A"
TX 4, I
```

Si se van colocando letras en la cola, realmente se irán colocando en las variables los valores de las letras.
Para enviar señales RS232 con un controlador de la serie MR-C3000 debe usar el comando "ETX"(transmisión de datos), en lugar del comando "TX" .

Ejemplo del comando
Lo siguiente es un ejemplo donde se lee el valor, tipo byte, del puerto "0" y se transmite constantemente vía RS232 a un terminal externo.

```
DIM   A   AS   BYTE


Main:
      A = BYTEIN(0)
      TX   4, A
      GOTO Main
```

MINIIN
Recibe señales Minibus a través del puerto minibus 2000

Sintaxis
MINIIN

Explicación del comando
Los datos miniBUS se reciben usando uno de los seis puertos miniBUS del controlador de la serie MR-C2000 mientras que el dato recibido no sea "0". Puede ver la estructura del programa aquí abajo.

```
DIM A AS BYTE

Retry:
      A = MINIIN
      IF A = 0 THEN GOTO Retry
```

MINIOUT
Envío de señales miniBUS por el puerto MiniBUS 2000

Sintaxis
MINIOUT [Datos], [Datos]...

Explicación del comando
Los datos miniBUS se transmiten usando el puerto miniBUS No. 6 del controlador de la serie MR-C2000. Los protocolos de comunicación Minibus son parecidos a los de RS232. Puede usar números, constantes, y variables para los [Datos]. Puede enviarse cualquier cantidad de datos, pero el número "0" NO puede transmitirse.

Ejemplo del comando
MINIOUT 100, 20, 76, 65

ERX
Recepción de señales RS232 por el puerto ERX 3000

Sintaxis
ERX [Velocidad], [variables de recepción], [Etiqueta rutina de recepción]

Explicación del comando
Los datos se reciben usando el puerto ERX del controlador de la serie MR-C3000 . Los códigos para la [velocidad del puerto] se indican en la tabla inferior.

Código	Configuración del puerto
2400	2400bps, 8Bit datos, Sin paridad, 1 bit Stop
4800	4800bps, 8Bit datos, Sin paridad, 1 bit Stop
9600	9600bps, 8Bit datos, Sin paridad, 1 bit Stop
14400	14400bps, 8Bit datos, Sin paridad, 1 bit Stop
19200	19200bps, 8Bit datos, Sin paridad, 1 bit Stop
28800	28800bps, 8Bit datos, Sin paridad, 1 bit Stop
38400	38400bps, 8Bit datos, Sin paridad, 1 bit Stop
57600	57600bps, 8Bit datos, Sin paridad, 1 bit Stop
76800	76800bps, 8Bit datos, Sin paridad, 1 bit Stop
115200	115200bps, 8Bit datos, Sin paridad, 1 bit Stop
230400	230400bps, 8Bit datos, Sin paridad, 1 bit Stop

[Variables de recepción] son las variables para recibir los datos, solo se pueden usar Variables declaradas como tipo byte.

[Etiqueta rutina de recepción] es la etiqueta del programa para la rutina de recepción.

```
Retry:
      ERX   9600,   A,      Retry
```

Para recibir señales RS232 desde un controlador de la serie MR-C2000, debe usar el comando "RX" en lugar del comando "ERX" (recepción de datos).

ETX
Transmisión de señales RS232 por el puerto ETX. 3000

Sintaxis
ETX [Velocidad del puerto], [Datos]

Explicación del comando
Los datos se transmiten a través del puerto ETX del controlador de la serie MR-C3000. Los códigos para la [velocidad del puerto] se muestran a continuación.

Código	Configuración del puerto
2400	2400bps, 8Bit datos, Sin paridad, 1 bit Stop
4800	4800bps, 8Bit datos, Sin paridad, 1 bit Stop
9600	9600bps, 8Bit datos, Sin paridad, 1 bit Stop
14400	14400bps, 8Bit datos, Sin paridad, 1 bit Stop
19200	19200bps, 8Bit datos, Sin paridad, 1 bit Stop
28800	28800bps, 8Bit datos, Sin paridad, 1 bit Stop
38400	38400bps, 8Bit datos, Sin paridad, 1 bit Stop
57600	57600bps, 8Bit datos, Sin paridad, 1 bit Stop
76800	76800bps, 8Bit datos, Sin paridad, 1 bit Stop
115200	115200bps, 8Bit datos, Sin paridad, 1 bit Stop
230400	230400bps, 8Bit datos, Sin paridad, 1 bit Stop

[Datos] es el valor a transmitir por el puerto ETX. Puede usar números, constantes y variables para [Datos]. Vea este ejemplo.

```
DIM I AS BYTE

I = "A"
ETX    9600, I
```

Si se va a transmitir la letra “A” debe enviar el código ASCII perteneciente a la letra “A”. Si quiere usar una variable, debe colocar el valor ASCII de la letra, antes de asignarla.

Para transmitir señales RS232 desde un controlador de la serie MR-C2000 debe usar el comando “TX”,en lugar del comando “ETX” (transmisión de datos).

Capítulo 11

roboBASIC

Procesado de señales analógicas

Descripción de los comandos

AD()

Una señal analógica del puerto AD es convertida a digital. 3000

Sintaxis
AD ([Puerto AD])

Explicación del comando
Existen ocho puertos AD , numerados del cero al siete, en los controladores de la serie MR-C3000 (Puertos I/O digitales 32 a 39), que convierten una señal analógica recibida desde un sensor o dispositivo externo en una señal digital. Puede usar constantes y variables tipo byte para definir el [puerto AD].

Ejemplo del comando
En el siguiente ejemplo, se envía un valor a un módulo LCD, después de recibir una señal analógica desde el puerto AD No. 1.

```
DIM a AS BYTE           Declara la variable byte “a”.
LCDINIT                 Se inicializa el módulo LCD.
CLS                     Se borran los datos del módulo LCD.
CSOFF                   El cursor desaparece.
MAIN:                   Se define una etiqueta llamada MAIN.
a = AD (1)              El valor recibido desde el puerto AD #1 se guarda en “a”.
LOCATE 5,0              El cursor se coloca en la posición 5.0 del LCD
PRINT FORMAT(a,DEC,2)   El valor recibido, a, se envía al módulo LCD como dos dígitos en notación decimal.
GOTO MAIN               Volver a MAIN.
```

REMOCON()

Lee datos desde el control remoto infrarrojo desde el puerto AD #7 3000

Sintaxis
REMOCON ([Remoto(#)])

Explicación del comando
El puerto No. 7 es usado para el control remoto infrarrojo. [Remoto(#)] tiene asignado el N. 1
, pero puede usar otros números dependiendo de la versión del controlador MR-C3000 (3024) usado. Para tener más detalles, consulte los ejemplos de los siguientes capítulos.

Remocon(1) IR remocon (tipo ID)

Ejemplo del comando

```
DIM a AS BYTE                 Declara la variable para los valores recibidos.
MAIN:Etiq.                    MAIN para recibir constantemente el valor del control remoto.
a = REMOCON(0)                El valor del control remoto se guarda en ”a”.
ON a GOTO MAIN,KEY1,KEY2,KEY3,KEY4   Volver a MAIN si no se recibe nada.
GOTO MAIN                     Volver a MAIN.
END
KEY1:                         Procesar cuando el valor recibido es 1.
 .....
GOTO MAIN                     Volver a MAIN
KEY2:                         Procesar cuando el valor recibido es 2.
 .....
GOTO MAIN                     Volver a MAIN
KEY3:                         Procesar cuando el valor recibido es 3.
 .....
GOTO MAIN                     Volver a MAIN
KEY4:                         Procesar cuando el valor recibido es 4.
 .....
GOTO MAIN                     Volver a MAIN
```

SONAR()

Lee la distancia calculada por un sensor de ultrasonidos conectado al puerto de ultrasonidos
3000

Sintaxis
SONAR ([Puerto de ultrasonidos])

Explicación del comando
Los puertos digitales I/O 0 a 23 del controlador de la serie MR-C3000 pueden ser utilizados como puertos de ultrasonidos 0 a 11. Vea la siguiente tabla.

Puerto digital I/O del controlador de la serie MR-C3000	Puerto de ultrasonidos
Puerto #0	#0 Puerto de salida ultrasónico
Puerto #1	#0 Puerto de entrada ultrasónico
Puerto #2	#1 Puerto de salida ultrasónico
Puerto #3	#1 Puerto de entrada ultrasónico
Puerto #4	#2 Puerto de salida ultrasónico
Puerto #5	#2 Puerto de entrada ultrasónico

Puerto #6	#3 Puerto de salida ultrasónico
Puerto #7	#3 Puerto de entrada ultrasónico
Puerto #8	#4 Puerto de salida ultrasónico
Puerto #9	#4 Puerto de entrada ultrasónico
Puerto #10	#5 Puerto de salida ultrasónico
Puerto #11	#5 Puerto de entrada ultrasónico
Puerto #12	#6 Puerto de salida ultrasónico
Puerto #13	#6 Puerto de entrada ultrasónico
Puerto #14	#7 Puerto de salida ultrasónico
Puerto #15	#7 Puerto de entrada ultrasónico
Puerto #16	#8 Puerto de salida ultrasónico
Puerto #17	#8 Puerto de entrada ultrasónico
Puerto #18	#9 Puerto de salida ultrasónico
Puerto #19	#9 Puerto de entrada ultrasónico
Puerto #20	#10 Puerto de salida ultrasónico
Puerto #21	#10 Puerto de entrada ultrasónico
Puerto #22	#11 Puerto de salida ultrasónico
Puerto #23	#11 Puerto de entrada ultrasónico

El valor del [puerto de ultrasonidos] debe ser un número fijo. El valor devuelto por el SONAR debe estar entre 0 y 3000. Si el valor devuelto es 0, la distancia no se ha medido, en cualquier otro caso, se devuelve un valor "XX".
Un sensor de ultrasonidos disponible para el controlador de la serie MR-C3000s es el modelo SRF04 de ROBOT ELECTRONICS Inc.

Ejemplo del comando

```
DIM A AS INTEGER        Declara la variable A.
A = SONAR(3)            El valor recibido (distancia) se guarda en la variable A
                        Se ha usado el puerto de ultrasonidos #3 (Puertos I/O#6, 7).
```

RCIN()

Lee el valor de los pulsos de emisoras y receptores RC 3000

Sintaxis

RCIN ([Puerto de recepción RC])

Explicación del comando

Los receptores RC se conectan al puerto de entrada AD de los controladores de la serie MR-C3000. Así, se pueden leer los valores recibidos desde la emisora. La tabla inferior muestra la configuración del [Puerto de recepción RC].

Puerto AD del controlador de la serie MR-C3000 (Digital In and Out Port Number)	Puerto recep. RC
Puerto #0 (Puerto #32)	Puerto recep. RC #0
Puerto #0 (Puerto #32)	Puerto recep. RC #1
Puerto #0 (Puerto #32)	Puerto recep. RC #2
Puerto #0 (Puerto #32)	Puerto recep. RC #3
Puerto #0 (Puerto #32)	Puerto recep. RC #4
Puerto #0 (Puerto #32)	Puerto recep. RC #5
Puerto #0 (Puerto #32)	Puerto recep. RC #6
Puerto #0 (Puerto #32)	Puerto recep. RC #7

Para evitar problemas causados por interferencias electrónicas, debe usar receptores FM en lugar de AM. Cuanto más avanzada sea la emisora, mas variadas serán las acciones y movimientos que pueda realizar.

Controlador
MR-C3000

Ejemplo de comando

```
DIM A AS BYTE
A = RCIN(0)        La señal recibida del puerto de recepción RC se almacena en la variable A.
```

GYRODIR

Define la dirección de los servos cuando son controlados por un giróscopo.3000

Sintaxis

GYRODIR [Grupo], [dirección del motor] …

Explicación del comando

Este proceso controla el sentido de giro de un grupo de servos cuando se conecta un giróscopo

a un puerto A/D de un controlador de la serie MR-C3000. El número máximo de giróscopos que puede conectar es 4. Vea el siguiente esquema.

Num. puerto AD (Núm. del puerto digital I/O del controlador de la serie MR-C3000.	Puerto giróscopo
Puerto #0 (Puerto #32)	Pto. de salida del canal Gyro #1
Puerto #0 (Puerto #33)	Pto. de salida del canal Gyro #2
Puerto #0 (Puerto #34)	Pto. de salida del canal Gyro #3
Puerto #0 (Puerto #35)	Pto. de salida del canal Gyro #4

Puerto #0 (Puerto #36)	Pto. de entrada del canal Gyro #1
Puerto #0 (Puerto #37)	Pto. de entrada del canal Gyro #2
Puerto #0 (Puerto #38)	Pto. de entrada del canal Gyro #3
Puerto #0 (Puerto #39)	Pto. de entrada del canal Gyro #4

Ya que un giróscopo es reversible, debe determinar el sentido del giro consultando el valor del servo. [Dirección del motor] es "0" o "1". Un valor 1 incrementa la posición del servo y 0 la disminuye.

Ejemplo del comando
GYRODIR G6A, 1, 1, 0, 0, 1, 0

GYROSET
Define que giróscopo controlará un grupo determinado de servos. 3000

Sintaxis
GYROSET [Grupo], [N Gyro] …

Explicación del comando
Gyroset determina cual de los servos en un [grupo] se controla por un giróscopo determinado.
[N Gyro] es el puerto del giróscopo que se usa para cada servo del grupo. Vea el ejemplo inferior.
01,02,03,04 : GWS PG03 11,12,13,14 : KRG-1
21,22,23,24 : reservado 31,32,33,34 : reservado

Ejemplo del comando

GYROSET G6B, 1, 1, 2, 2, 0, 0
servo #6 recibe el sensor del giróscopo # r1 y lo procesa
servo #7 recibe el sensor del giróscopo # r1 y lo procesa
servo #8 recibe el sensor del giróscopo # r2 y lo procesa
servo #9 recibe el sensor del giróscopo # r2 y lo procesa
servos #10 y #11 no usan el sensor del giróscopo .

GYROSENSE
Define la sensibilidad del servo a un giróscopo. 3000

Sintaxis
GYROSENSE [Grupo], [sensibilidad N Gyro] …

Explicación del comando
Se pueden conectar 4 giróscopos a un controlador de la serie MR-C3000. GYROSENSE define la sensibilidad de un servo individual a un giróscopo.
[sensibilidad N Gyro] usa números entre 0 y 255 o constantes para controlar la sensibilidad de cada servo en un grupo. Un valor de "0" no cambia la sensibilidad de los servos. A medida que el número aumenta, la reacción del servo respecto al giro aumenta también.

Ejemplo del comando
GYROSENSE G6A, 100, 100, 255, 255, 50, 50
Servos #0 y #1 se configuran con sensibilidad 100 del Gyro.
Servos #2 y #3 configurados con la sensibilidad máxima (255) del gyro.
Servos #4 y #5 se configuran con sensibilidad 50 del giro.

Capítulo 12

roboBasic

Comandos de control de procesos, y otros

ON…GOTO
Salto condicional dependiendo del valor de una variable.

Sintaxis
ON [Variable] GOTO [etiqueta], [etiqueta]...

Ejemplo del comando
"ON… GOTO" es el comando que le permitirá saltar a rutinas determinadas del programa dependiendo del valor de una [Variable]. Si usa el comando IF, puede seguir usando el comando On…GOTO. De hecho, si comparamos los comandos, ON…GOTO permite creación de programas más pequeños.

Lo siguientes es una comparación entre IF y ON… GOTO.

Siguiendo al comando GOTO viene [etiqueta]. [etiqueta] se incrementa siempre que

la [variable] sea cierta. Puede usar números y constantes para la [variable]. El valor máximo de [etiqueta] es 255.

```
A = BYTEIN(0)

IF A = 0 THEN
   GOTO 10
ELSEIF A = 1 THEN
   GOTO 20
ELSEIF A = 2 THEN
   GOTO 30
ELSE
   GOTO 40
ENDIF
```

```
A = BYTEIN(0)

ON A GOTO 10, 20, 30
IF A>2 THEN GOTO 40
```

RND
Aleatorio.

Sintaxis
RND

Explicación del comando
Use el comando RND para crear un valor aleatorio con los controladores de la serie. Este comando, creará un valor aleatorio entre 0 y 255.

Ejemplo del comando
DIM A AS BYTE

```
A = RND
BYTEOUT 0, A          Se envía un valor aleatorio en formato byte al puerto 0.
```

REMARK
Coloca un comentario dentro del código

Sintaxis
REMARK [Descripción]

Explicación del comando
Es una buena práctica de programación, el colocar comentarios dentro del código para explicar determinadas funciones. Esto se hace con el carácter (') o con el comando REMARK. Siguiendo al comando, se coloca el comentario en la misma línea. REMARK no causa ningún efecto en la ejecución del programa.

Ejemplo del comando
```
REMARK se encienden 8 LEDs
BYTEOUT 0, 0
```

Capítulo 13

roboBasic

Descripción de comandos

'$DEVICE
Define el controlador a usar por el programa.

Sintaxis
'$DEVICE [Controlador]

Explicación del comando
Este comando define para que controlador se va a compilar un programa para su posterior volcado. Si ya se había asignado un controlador, se cambiará al especificado en [controlador].

Ejemplo del comando
```
'$DEVICE    MRC2000    El programa usará un controlador MR-C2000.
```

'$LIMIT
Limita el rango de movimientos de cada servo. 3000

Sintaxis
'$LIMIT [Número de motor, [valor mínimo], [valor máximo]

Explicación del comando
El comando "$LIMIT" se utiliza para limitar los recorridos totales de los servos, y así prevenir posibles daños causados al forzar sus límites.

[Número de motor] especifica el servo. El rango oscila entre 0 y 31. [Valor mínimo] y [Valor máximo] son los ángulos

máximos de recorridos deseados entre 10 y 190. Los valores de los ángulos recorridos por un servo son 10 a 190 por defecto.

Ejemplo del comando
```
'$LIMIT     0, 50, 100    Se limita el ángulo del servo entre 50 y 100.
```

Manual de Instrucciòn

ROBONOVA-I

www.hitecrobotics.com

MULTIPLEX Modellsport GmbH & Co. KG
www.robonova.de

Castellano V1.00